Teoria della Computabilità e Struttura dei Gradi di Turing

Prolegomeni: La Nascita della Computabilità come Disciplina Matematica

La teoria della computabilità emerge negli anni '30 del ventesimo secolo da una confluenza di sviluppi in logica matematica, metamatematica e filosofia della matematica. Il contesto intellettuale è dominato dal programma di Hilbert, che cerca di fondare l'intera matematica su basi formali e dimostrabili. La questione dell'Entscheidungsproblem - esiste una procedura meccanica per decidere la verità di qualsiasi enunciato matematico? - rappresenta uno dei problemi centrali del programma hilbertiano.

Ma cosa significa esattamente "procedura meccanica"? Prima degli anni '30, questa nozione rimaneva intuitiva e informale. Matematici e logici parlavano di "algoritmi" e "metodi effettivi" senza una definizione precisa. La grande conquista concettuale di questo periodo fu la formalizzazione matematica rigorosa della nozione di computabilità.

Church propose nel 1936 il λ-calcolo come formalismo per le funzioni calcolabili. Il λ-calcolo è un sistema di riscrittura simbolica basato su astrazione funzionale e applicazione. Una funzione nel λ-calcolo è un termine della forma $\lambda x.M$, dove M è un termine che può contenere x. L'applicazione di $\lambda x.M$ a un argomento N produce $M[N/x]$, la sostituzione di N per x in M. Church dimostrò che molte funzioni matematiche naturali sono definibili in questo formalismo e propose che le funzioni λ-definibili catturino esattamente la nozione intuitiva di calcolabilità.

Contemporaneamente, Turing introdusse nel 1936 un modello completamente diverso: le macchine automatiche, oggi chiamate macchine di Turing. Una macchina di Turing consiste di un nastro infinito diviso in celle, una testina di lettura/scrittura che può muoversi lungo il nastro, e un controllo finito che determina l'azione della macchina basandosi sullo stato corrente e sul simbolo letto. La macchina può scrivere simboli, muovere la testina a sinistra o destra, e cambiare stato. Una computazione è una sequenza di configurazioni (stato, contenuto del nastro, posizione della testina) che evolve secondo le regole della macchina.

Il modello di Turing ha un'intuizione fisica diretta: rappresenta un agente che esegue operazioni elementari su simboli, con memoria illimitata ma accessibile solo localmente. Questa concretezza fisica del modello di Turing contrasta con l'astrattezza algebrica del λ-calcolo, eppure i due formalismi si rivelano equivalenti in potere espressivo.

Kleene sviluppò nel 1936 la teoria delle funzioni ricorsive, basata su una costruzione algebrica. Le funzioni ricorsive sono generate da funzioni iniziali elementari (zero, successore, proiezioni) attraverso operazioni di composizione, ricorsione primitiva, e minimizzazione illimitata (l'operatore μ). Le funzioni ricorsive primitive sono quelle ottenibili

senza l'operatore μ e corrispondono a computazioni che terminano sempre. Le funzioni ricorsive generali includono anche l'operatore μ, che cerca il minimo numero soddisfacente una condizione, e possono non terminare.

Gödel aveva già esplorato nel 1931 una classe ristretta di funzioni ricorsive primitive nel contesto del suo teorema di incompletezza. La numerazione di Gödel, che assegna numeri naturali a formule e dimostrazioni, richiede funzioni che sono effettivamente calcolabili ma cui Gödel non aveva dato una caratterizzazione generale.

Post propose nel 1936 i sistemi di produzione (Post production systems), sistemi di riscrittura di stringhe che generalizzano le grammatiche formali. Un sistema di Post consiste di regole che trasformano stringhe in altre stringhe. Post dimostrò che questi sistemi hanno lo stesso potere espressivo delle macchine di Turing.

La convergenza di questi formalismi apparentemente diversi verso la stessa classe di funzioni è uno dei fenomeni più notevoli della logica matematica. Church, Turing, Kleene e Post dimostrarono indipendentemente che le loro nozioni di computabilità coincidono: una funzione è λ-definibile se e solo se è Turing-calcolabile se e solo se è ricorsiva generale se e solo se è computabile da un sistema di Post. Questa convergenza fornisce una forte evidenza per la tesi di Church-Turing.

La tesi di Church-Turing afferma che ogni funzione "effettivamente calcolabile" nel senso intuitivo è ricorsiva generale (o equivalentemente, Turing-calcolabile, λ-definibile, etc.). Questa non è una proposizione matematica formale, perché un lato dell'equivalenza (la nozione intuitiva di calcolabilità effettiva) non è un concetto matematico preciso. La tesi è piuttosto una identificazione filosofica tra un concetto informale e uno formale. Non può essere dimostrata in senso stretto, ma solo argomentata attraverso l'accumulo di evidenze: la convergenza dei vari formalismi, il fatto che tutte le funzioni intuitivamente calcolabili finora considerate si rivelano essere ricorsive, l'assenza di controesempi plausibili dopo novant'anni di ricerca.

La tesi di Church-Turing ha conseguenze profonde per la filosofia della matematica e della mente. Implica che non esistono "metodi di calcolo nascosti" che superano le macchine di Turing. Ogni calcolo che un matematico può eseguire in linea di principio può essere eseguito da una macchina di Turing. Questo ha implicazioni per il dibattito su mente e macchina: se la mente umana è computazionale, allora è equivalente a una macchina di Turing in potere di calcolo.

Tuttavia, la tesi di Church-Turing è puramente qualitativa: riguarda la distinzione calcolabile/non-calcolabile, non la distinzione veloce/lento. Una funzione può essere Turing-calcolabile ma richiedere tempo esponenziale, doppiamente esponenziale, o addirittura non-elementare. La teoria della complessità computazionale, sviluppata negli anni '60 e '70, studia queste distinzioni quantitative. La teoria dei gradi di Turing, invece, astrae completamente da considerazioni di complessità temporale o spaziale e si concentra sulla gerarchia qualitativa della calcolabilità.

Riducibilità di Turing: Fondamenti e Analisi Profonda

La nozione di riducibilità di Turing, introdotta da Turing nel 1939 in un articolo fondamentale sui "sistemi di logica basati sugli ordinali", rappresenta uno degli strumenti concettuali più potenti della logica matematica contemporanea. Questa nozione permette di comparare la difficoltà computazionale di problemi anche quando entrambi sono indecidibili, aprendo la possibilità di una teoria graduata della non-calcolabilità.

Un insieme A di numeri naturali è Turing-riducibile a un insieme B, denotato $A \leq_T B$, quando esiste una macchina di Turing M che computa la funzione caratteristica χ_A di A (dove $\chi_A(n) = 1$ se $n \in A$ e $\chi_A(n) = 0$ altrimenti) avendo accesso a un oracolo per B. L'oracolo è un dispositivo ideale, una "black box", che, interrogato su un numero n, risponde immediatamente e infallibilmente se n appartiene a B oppure no.

Analizziamo questa definizione con precisione tecnica. Una macchina di Turing con oracolo per B è una macchina di Turing ordinaria equipaggiata con tre nastri speciali: il nastro query, il nastro answer, e un stato speciale query. Quando la macchina entra nello stato query, il contenuto del nastro query (interpretato come un numero naturale n) viene "letto" dall'oracolo, che scrive sul nastro answer il valore di $\chi_B(n)$, e la macchina transisce in uno stato appropriato per continuare la computazione. Questa operazione avviene in un singolo passo computazionale, senza consumare tempo: l'oracolo è infinitamente veloce.

La definizione assume che l'oracolo sia consultabile su qualsiasi numero naturale e che risponda sempre correttamente. Non ci sono restrizioni sul numero di consultazioni: una computazione può interrogare l'oracolo una volta, dieci volte, o infinitamente molte volte (sebbene, naturalmente, una computazione che termina in tempo finito possa fare solo finitamente molte query). Questo distingue la riducibilità di Turing dalla riducibilità truth-table, che descriveremo tra breve.

La potenza espressiva della riducibilità di Turing emerge quando consideriamo che la riduzione da A a B può dipendere adattivamente dalle risposte dell'oracolo. Data una query n_1, riceviamo una risposta $r_1 \in \{0,1\}$. La prossima query n_2 può dipendere da r_1. La query n_3 può dipendere da r_1 e r_2. E così via. Questa adattività permette strategie di riduzione molto più flessibili rispetto a riducibilità non-adaptive come la riducibilità truth-table.

Per rendere concrete queste idee, consideriamo alcuni esempi fondamentali. Sia $K = \{e : \Phi_e(e) \text{ converge}\}$ il problema dell'arresto (l'insieme degli indici di macchine di Turing che si fermano sui propri indici). Sia $\bar{K}$ il complemento di K. Allora $K \equiv_T \bar{K}$: gli insiemi sono Turing-equivalenti. Dato un oracolo per K, posso decidere $\bar{K}$ semplicemente negando le risposte dell'oracolo. Conversamente, dato un oracolo per $\bar{K}$, posso decidere K negando le risposte. Questa equivalenza è banale ma illuminante: mostra che "decidere" un insieme e "decidere" il suo complemento hanno la stessa difficoltà dal punto di vista della riducibilità di Turing.

Consideriamo ora $Tot = \{e : \Phi_e \text{ è totale}\}$, l'insieme degli indici di funzioni ricorsive totali. Allora $K <_T Tot$: K è Turing-riducibile a Tot ma Tot non è Turing-riducibile a K. Per vedere che $K \leq_T Tot$, osserviamo che $e \in K$ se e solo se esiste una macchina totale che, su input 0, simula $\Phi_e(e)$ e si ferma se e solo se $\Phi_e(e)$ converge. Possiamo costruire ricorsivamente (senza oracolo) l'indice i di una tale macchina da e, e poi $e \in K$ se e solo se $i \in Tot$. Questa riduzione è in realtà una riducibilità many-one, come vedremo.

Per vedere che Tot $\not\leq_T K$, supponiamo per assurdo che esista una macchina M con oracolo per K che decide Tot. Consideriamo la seguente strategia diagonale. Definiamo ricorsivamente una macchina N il cui comportamento dipende da M. Su input n, N simula M sull'indice di N stesso. Se M (con oracolo per K) decide che N è totale, allora N entra in un loop infinito su input n. Se M decide che N non è totale, allora N si ferma immediatamente su input n. Ora, se N è totale, allora per costruzione M decide che N è totale, ma allora N entra in loop su qualche input, contraddizione. Se N non è totale, allora M decide che N non è totale, ma allora N si ferma su tutti gli input, contraddizione. Questa diagonalizzazione mostra che nessuna macchina con oracolo per K può decidere Tot.

La riducibilità di Turing soddisfa proprietà naturali di una preordine:

Riflessività: $A \leq_T A$ per ogni A. Questo è ovvio: una macchina con oracolo per A può decidere A semplicemente consultando l'oracolo.

Transitività: Se $A \leq_T B$ e $B \leq_T C$, allora $A \leq_T C$. Dimostrazione: sia M_1 una macchina con oracolo per B che computa χ_A, e sia M_2 una macchina con oracolo per C che computa χ_B. Costruiamo una macchina M con oracolo per C che computa χ_A simulando M_1. Ogni volta che M_1 consulta l'oracolo per B su input n, invece di ottenere la risposta direttamente, simuliamo M_2 con oracolo per C su input n per ottenere $\chi_B(n)$. Questa simulazione può richiedere consultazioni multiple dell'oracolo per C, ma alla fine produce $\chi_B(n)$, permettendo a M_1 di continuare. La composizione di queste simulazioni computa χ_A con oracolo per C.

La transitività richiede una verifica accurata che la simulazione di M_2 all'interno di M_1 non crei problemi di divergenza. Se M_2 su qualche input diverge, allora la simulazione di M_1 diverge in quel punto. Ma questo è accettabile: stiamo costruendo una macchina parziale con oracolo che computa χ_A sui valori dove entrambe M_1 e le sue chiamate a M_2 convergono. Poiché M_1 con oracolo corretto per B computa χ_A totalmente, e M_2 con oracolo corretto per C computa χ_B totalmente, la composizione computa χ_A totalmente.

La riducibilità di Turing non è antisimmetrica: possono esistere A e B distinti tali che $A \leq_T B$ e $B \leq_T A$. L'esempio più semplice è A e $\bar{A}$: sono distinti (assumendo A non vuoto e non uguale a $\mathbb{N}$), ma $A \leq_T \bar{A}$ e $\bar{A} \leq_T A$. Per ottenere una relazione d'ordine, introduciamo l'equivalenza di Turing.

Gradi di Turing: Costruzione e Proprietà Fondamentali

Due insiemi A e B sono Turing-equivalenti, denotato $A \equiv_T B$, quando simultaneamente $A \leq_T B$ e $B \leq_T A$. Questa relazione è chiaramente un'equivalenza:

Riflessività: $A \equiv_T A$ perché $A \leq_T A$.

Simmetria: Se $A \equiv_T B$ allora $B \equiv_T A$ per definizione simmetrica.

Transitività: Se $A \equiv_T B$ e $B \equiv_T C$, allora $A \leq_T B \leq_T C$ implica $A \leq_T C$ per transitività di $\leq_T$, e similmente $C \leq_T B \leq_T A$ implica $C \leq_T A$, quindi $A \equiv_T C$.

Le classi di equivalenza rispetto a ≡_T sono chiamate gradi di Turing. Il grado di un insieme A, che denotiamo deg(A) o semplicemente **a** quando il rappresentante specifico è irrilevante, raccoglie tutti gli insiemi che hanno esattamente la stessa difficoltà computazionale di A nel senso della riducibilità di Turing. Formalmente:

$\deg(A) = \{B \subseteq \mathbb{N} : B \equiv_T A\}$

Ogni insieme appartiene a esattamente un grado. La collezione di tutti i gradi forma l'insieme D, che chiamiamo la struttura dei gradi di Turing.

Sulla collezione D possiamo definire un ordinamento naturale. Per gradi **a** e **b**, definiamo:

a ≤ **b** ⇔ esiste $A \in \mathbf{a}$ e $B \in \mathbf{b}$ tale che $A \leq_T B$

Dobbiamo verificare che questa definizione sia ben posta, cioè che non dipenda dalla scelta dei rappresentanti A e B. Supponiamo $A_1 \equiv_T A_2$ e $B_1 \equiv_T B_2$, e che $A_1 \leq_T B_1$. Vogliamo mostrare $A_2 \leq_T B_2$. Per ipotesi:

- $A_2 \leq_T A_1$ (da $A_1 \equiv_T A_2$)
- $A_1 \leq_T B_1$ (per ipotesi)
- $B_1 \leq_T B_2$ (da $B_1 \equiv_T B_2$)

Per transitività di ≤_T, otteniamo $A_2 \leq_T B_2$. Quindi l'ordinamento sui gradi è ben definito.

La struttura (D, ≤) forma un ordine parziale superiore:

Riflessività: **a** ≤ **a** per ogni **a**, perché prendendo $A \in \mathbf{a}$, abbiamo $A \leq_T A$.

Antisimmetria: Se **a** ≤ **b** e **b** ≤ **a**, allora **a** = **b**. Questo segue dalla definizione: se **a** ≤ **b**, esistono $A \in \mathbf{a}$ e $B \in \mathbf{b}$ con $A \leq_T B$. Se **b** ≤ **a**, esistono $B' \in \mathbf{b}$ e $A' \in \mathbf{a}$ con $B' \leq_T A'$. Ma $B \equiv_T B'$ e $A \equiv_T A'$, quindi $A \leq_T B \leq_T B' \leq_T A' \leq_T A$, da cui $A \equiv_T B$ e quindi **a** = **b**.

Transitività: Se **a** ≤ **b** e **b** ≤ **c**, allora **a** ≤ **c**. Questo segue dalla transitività di ≤_T.

La struttura D ha un elemento minimo, che denotiamo **0** (o talvolta **0'** nella letteratura). Questo è il grado contenente tutti gli insiemi ricorsivi. Verifichiamo che tutti gli insiemi ricorsivi sono mutuamente Turing-equivalenti. Se A e B sono entrambi ricorsivi, posso decidere A senza consultare alcun oracolo, quindi certamente posso decidere A avendo un oracolo per B (semplicemente ignoro l'oracolo). Simmetricamente, posso decidere B avendo un oracolo per A. Quindi $A \equiv_T B$.

Il grado **0** contiene tutti e soli gli insiemi decidibili: ∅, ℕ, {0}, {1, 3, 5}, e infiniti altri. Dal punto di vista della riducibilità di Turing, tutti questi insiemi sono equivalenti. Questo può sembrare contro-intuitivo: certamente {1, 3, 5} è più "complesso" di ∅ in qualche senso. Ma dal punto di vista della Turing-riducibilità, che misura solo la capacità di decidere appartenenza, non la complessità temporale o strutturale, questi insiemi sono indistinguibili.

Questa osservazione rivela una scelta metodologica fondamentale della teoria dei gradi: stiamo astraendo completamente da considerazioni di complessità concreta. Non ci importa se decidere A richiede tempo O(n), $O(2^n)$, o O(2^(2^n)). Ci importa solo se A può essere

deciso in tempo finito usando una procedura algoritmica. Questa astrazione radicale permette di concentrarsi sulla struttura qualitativa della calcolabilità, ma richiede teorìe separate (teoria della complessità computazionale) per studiare distinzioni quantitative.

La struttura D non ha elemento massimo. Questo sarà conseguenza dell'esistenza del salto di Turing, che introdurremo in dettaglio. Per ora, anticipiamo che per ogni insieme A possiamo costruire un insieme A' che è strettamente più difficile computazionalmente: A <_T A'. Iterando questa costruzione indefinitamente, otteniamo una catena strettamente crescente senza limite superiore.

Operazioni sui Gradi: Join, Meet e Salto

La struttura D, essendo un ordine parziale, solleva naturalmente la questione dell'esistenza di suprema (least upper bounds) e infima (greatest lower bounds) per coppie di elementi. Consideriamo prima i suprema.

Dati due insiemi A e B, definiamo la loro somma diretta (o join) $A \oplus B$ come:

$A \oplus B = \{2n : n \in A\} \cup \{2n+1 : n \in B\}$

In altre parole, $A \oplus B$ codifica simultaneamente A e B usando numeri pari per A e numeri dispari per B. È immediato verificare che:

1. $A \leq_T A \oplus B$: dato un oracolo per $A \oplus B$, posso decidere $n \in A$ consultando l'oracolo su $2n$.

2. $B \leq_T A \oplus B$: simmetricamente, posso decidere $n \in B$ consultando l'oracolo su $2n+1$.

3. $A \oplus B$ è il minimo maggiorante comune: se $C \geq_T A$ e $C \geq_T B$, allora $C \geq_T A \oplus B$.

La terza proprietà richiede dimostrazione. Supponiamo di avere macchine M_A e M_B con oracolo per C che computano rispettivamente χ_A e χ_B. Costruiamo una macchina M con oracolo per C che computa $\chi_{(A \oplus B)}$. Su input n, M determina se n è pari o dispari. Se $n = 2m$ è pari, M simula M_A su input m con oracolo per C, ottenendo $\chi_A(m) = \chi_{(A \oplus B)}(n)$. Se $n = 2m+1$ è dispari, M simula M_B su input m con oracolo per C, ottenendo $\chi_B(m) = \chi_{(A \oplus B)}(n)$. Quindi $A \oplus B \leq_T C$.

Questo dimostra che $A \oplus B$ è il supremum di A e B nella struttura $(D, \leq_T)$. Possiamo quindi definire il join di due gradi:

$\mathbf{a} \vee \mathbf{b} = \deg(A \oplus B)$ dove $A \in \mathbf{a}$ e $B \in \mathbf{b}$

Dobbiamo verificare che questa definizione non dipende dalla scelta dei rappresentanti. Se $A_1 \equiv_T A_2$ e $B_1 \equiv_T B_2$, allora dobbiamo mostrare $A_1 \oplus B_1 \equiv_T A_2 \oplus B_2$.

$A_1 \oplus B_1 \leq_T A_2 \oplus B_2$: Dato un oracolo per $A_2 \oplus B_2$, possiamo computare A_2 (consultando su numeri pari) e B_2 (consultando su numeri dispari). Poiché $A_1 \leq_T A_2$ e $B_1 \leq_T B_2$, possiamo computare A_1 e B_1, e quindi $A_1 \oplus B_1$.

La direzione opposta $A_2 \oplus B_2 \leq_T A_1 \oplus B_1$ è simmetrica. Quindi il join sui gradi è ben definito.

L'operazione $\vee$ fa di D un upper semilattice: ogni coppia di elementi ha un supremum. Le proprietà di $\vee$ includono:

Commutatività: $\mathbf{a} \vee \mathbf{b} = \mathbf{b} \vee \mathbf{a}$, perché $A \oplus B \equiv_T B \oplus A$ (possiamo codificare la somma con pari/dispari in ordine inverso attraverso una permutazione ricorsiva).

Associatività: $(\mathbf{a} \vee \mathbf{b}) \vee \mathbf{c} = \mathbf{a} \vee (\mathbf{b} \vee \mathbf{c})$, che riflette il fatto che somme dirette multiple sono associative a meno di ricodifica ricorsiva.

Idempotenza: $\mathbf{a} \vee \mathbf{a} = \mathbf{a}$, perché $A \oplus A \equiv_T A$.

Elemento neutro: $\mathbf{a} \vee \mathbf{0} = \mathbf{a}$, perché se R è ricorsivo, $A \oplus R \equiv_T A$.

La questione del meet (infimum) è molto più delicata. Non è vero in generale che ogni coppia di gradi abbia un infimum in D. Kleene e Post congetturarono negli anni '40 che ogni coppia di gradi avesse un infimum, ma questa congettura fu confutata negli anni '60.

Teorema (Lachlan, Yates): Esistono gradi **a** e **b** senza infimum in D.

La dimostrazione è altamente non-triviale e usa costruzioni sofisticate con il metodo delle priorità. L'idea è costruire due gradi **a** e **b** e simultaneamente diagonalizzare contro ogni possibile candidato per l'infimum. Se **c** fosse l'infimum di **a** e **b**, dovrebbe soddisfare:

1. $\mathbf{c} \leq \mathbf{a}$ e $\mathbf{c} \leq \mathbf{b}$
2. Per ogni **d**, se $\mathbf{d} \leq \mathbf{a}$ e $\mathbf{d} \leq \mathbf{b}$, allora $\mathbf{d} \leq \mathbf{c}$

La costruzione assicura che per ogni grado $\mathbf{c} \leq \mathbf{a}, \mathbf{b}$, esiste un grado **d** strettamente maggiore di **c** che è ancora $\leq \mathbf{a}, \mathbf{b}$. Questo impedisce a qualsiasi **c** di essere il massimo lower bound.

L'inesistenza di infima in generale ha conseguenze strutturali profonde. D non è un lattice, solo un upper semilattice. Questo distingue D da molte strutture algebriche naturali in matematica, che tipicamente sono lattices completi o almeno lattices. La "incompletezza" di D riflette la complessità intrinseca della struttura computazionale.

Il Metodo delle Priorità: Analisi Tecnica Approfondita

Il metodo delle priorità, introdotto da Friedberg e Muchnik nel 1956-1957, è diventato lo strumento più potente e versatile nella teoria della computabilità per costruire insiemi con proprietà specificate. Il metodo è essenzialmente una tecnica di diagonalizzazione infinita organizzata con un sistema di priorità che risolve conflitti tra requisiti potenzialmente incompatibili.

Descriviamo il metodo in dettaglio attraverso la dimostrazione del teorema di Friedberg-Muchnik: esistono gradi r.e. incomparabili. Vogliamo costruire due insiemi r.e. A e B tali che:

- $A \not\leq_T B$
- $B \not\leq_T A$
- A e B non sono ricorsivi

Per garantire $A \not\leq_T B$, dobbiamo assicurare che nessuna macchina di Turing con oracolo per B computi la funzione caratteristica di A. Formalmente, per ogni indice e, dobbiamo soddisfare il requisito:

$N_e: \Phi_e^B \neq \chi_A$

dove Φ_e^B denota la funzione (parziale) computata dalla e-esima macchina di Turing con oracolo per B.

Per soddisfare N_e, usiamo una strategia diagonale. Associamo a N_e un testimone (witness) x_e. La strategia per N_e è:

1. Inizializza x_e a un valore "fresco" non usato da requisiti di priorità più alta.

2. Aspetta finché $\Phi_e^{B_s}(x_e)$ converge a qualche valore v, dove B_s denota l'approssimazione finita di B allo stadio s della costruzione.

3. Quando $\Phi_e^{B_s}(x_e) = v$ converge:

    - Se v = 0, enumera x_e in A, così $\chi_A(x_e) = 1 \neq 0 = \Phi_e^{B_s}(x_e)$
    - Se v = 1, lascia x_e fuori da A, così $\chi_A(x_e) = 0 \neq 1 = \Phi_e^{B_s}(x_e)$
4. Assicurati che la convergenza $\Phi_e^{B_s}(x_e) = v$ rimanga valida per tutti gli stadi futuri, cioè che B non cambi sui numeri consultati durante questa computazione.

Il punto 4 è cruciale e problematico. Durante la computazione di $\Phi_e^{B_s}(x_e)$, la macchina consulta l'oracolo per B su un insieme finito di numeri, chiamiamo questo insieme U_e (la use set). La computazione $\Phi_e^{B_s}(x_e) = v$ è valida finché $B \upharpoonright U_e$ (la restrizione di B a U_e) non cambia. Se successivamente enumeriamo qualche numero in U_e in B, la computazione viene "invalidata": $\Phi_e^B(x_e)$ potrebbe ora divergere o convergere a un valore diverso.

Questo crea un conflitto potenziale. Supponiamo che al tempo s soddisfiamo N_e mettendo x_e in A basandoci su $\Phi_e^{B_s}(x_e) = 0$. Successivamente, al tempo $t > s$, potremmo dover soddisfare un altro requisito M_j, e la strategia per M_j potrebbe richiedere di enumerare qualche numero $y \in U_e$ in B. Questo invaliderebbe la computazione $\Phi_e^{B_s}(x_e)$, e il requisito N_e non sarebbe più soddisfatto: $\Phi_e^B(x_e)$ potrebbe ora divergere o convergere a $1 = \chi_A(x_e)$.

Il sistema di priorità risolve questo conflitto assegnando priorità ai requisiti. Elenchiamo tutti i requisiti in un ordine lineare:

$N_0, M_0, N_1, M_1, N_2, M_2, N_3, M_3, ...$

dove M_e è il requisito simmetrico che garantisce B $\not\leq_T$ A. I requisiti sono ordinati per priorità decrescente: N_0 ha priorità massima, poi M_0, poi N_1, e così via. La regola fondamentale è: un requisito di priorità più alta può danneggiare (injure) un requisito di priorità più bassa, ma non viceversa.

Concretamente, questo significa: se al tempo s abbiamo soddisfatto N_e e successivamente un requisito di priorità più alta (diciamo N_j con j < e o M_j con j ≤ e) richiede di enumerare un numero in B che invalida la computazione usata per soddisfare N_e, allora permettiamo questa enumerazione. N_e viene "danneggiato" e dovrà essere risoddisfatto più tardi. Viceversa, se un requisito di priorità più bassa vorrebbe enumerare un numero che invaliderebbe N_e, lo preveniamo: proteggiamo N_e a scapito del requisito di priorità più bassa.

La costruzione procede per stadi s = 0, 1, 2, ... A ogni stadio s, abbiamo approssimazioni finite A_s e B_s degli insiemi finali A e B. Inizialmente A_0 = B_0 = ∅. Al stadio s, consideriamo il requisito di priorità più alta che non è ancora soddisfatto definitivamente e tentiamo di soddisfarlo.

Per determinare se N_e è "soddisfatto" allo stadio s, verifichiamo:

- Esiste un testimone x_e associato a N_e?
- Se sì, la computazione $\Phi_e^{B_s}(x_e)$ è divergente, oppure converge a un valore diverso da $\chi_{A_s}(x_e)$?

Se una di queste condizioni è vera, N_e è "correntemente soddisfatto" allo stadio s. Ma questo non significa che N_e sia soddisfatto definitivamente: potrebbe essere danneggiato da requisiti di priorità più alta in stadi futuri.

L'algoritmo della costruzione al stadio s opera come segue:

Input: Approssimazioni A_s, B_s costruite finora.

Processo:

1. Per ogni requisito R in ordine di priorità: a. Se R = N_e e N_e non è correntemente soddisfatto:
    - Se non esiste testimone x_e, scegli x_e fresco (maggiore di tutti i numeri menzionati finora)
    - Aspetta che $\Phi_e^{B_s}(x_e)$ converga (se non converge, passa al prossimo requisito)
    - Se $\Phi_e^{B_s}(x_e) = v$ converge con $v \neq \chi_{A_s}(x_e)$:
        - Mantieni lo stato corrente (N_e già soddisfatto)
    - Se $\Phi_e^{B_s}(x_e) = \chi_{A_s}(x_e)$:
        - Cambia χ_A su x_e: enumera x_e in A_{s+1} se v = 0, oppure garantisci $x_e \notin A$ se v = 1
        - Segna N_e come "soddisfatto allo stadio s"
        - Registra U_e = insieme dei numeri consultati in B durante $\Phi_e^{B_s}(x_e)$

2. b. Se R = M_e, simmetricamente per garantire B ⩽_T A

3. Passa allo stadio s+1

Output: A_{s+1}, B_{s+1}

Questa descrizione è ancora incompleta perché non specifica come "proteggere" le use sets contro danneggiamenti da requisiti di priorità più bassa. La versione completa richiede che quando enumeriamo un numero in A o B, verifichiamo se questo numero appare in qualche use set di un requisito di priorità più alta già soddisfatto. Se sì, preveniamo l'enumerazione.

La dimostrazione che questa costruzione funziona richiede provare due lemmi fondamentali:

Lemma 1 (Finitezza dei Danneggiamenti): Ogni requisito N_e viene danneggiato solo finitamente molte volte.

Dimostrazione per Induzione su e:

- Base: N_0 ha priorità massima, quindi non può essere danneggiato da nessun requisito. Una volta soddisfatto N_0, rimane soddisfatto per sempre.
- Passo Induttivo: Supponiamo che tutti i requisiti di priorità più alta di N_e siano eventualmente soddisfatti definitivamente e non vengano più danneggiati dopo un certo stadio s_e. Dopo s_e, questi requisiti non enumerano più numeri in B che potrebbero invalidare le computazioni di N_e. L'unica fonte di danneggiamento per N_e sono i requisiti di priorità più alta, che dopo s_e sono stabili. Quindi N_e può essere danneggiato solo finitamente molte volte: al più tante volte quante il numero totale di enumerazioni in B da parte di requisiti di priorità più alta prima dello stadio s_e, che è finito.

Lemma 2 (Soddisfacimento Definitivo): Ogni requisito N_e viene eventualmente soddisfatto definitivamente.

Dimostrazione: Per il Lemma 1, esiste uno stadio s_e dopo il quale N_e non viene più danneggiato. Se N_e non è soddisfatto allo stadio s_e, allora a qualche stadio s > s_e la costruzione tenterà di soddisfare N_e. Ci sono due possibilità:

Caso 1: Φ_e^B(x_e) diverge. Allora N_e è soddisfatto perché Φ_e^B non è totale, quindi Φ_e^B ≠ χ_A.

Caso 2: Φ_e^B(x_e) converge a qualche valore v. Allora al primo stadio t ≥ s_e dove Φ_e^{B_t}(x_e) converge, la costruzione enumera x_e in A se v = 0, o garantisce x_e ∉ A se v = 1, creando una differenza tra Φ_e^B(x_e) e χ_A(x_e). Poiché dopo s_e nessun requisito di priorità più alta danneggia N_e, la computazione Φ_e^{B_t}(x_e) = v rimane valida per tutti gli stadi futuri (B non cambia sulla use set U_e), e quindi Φ_e^B(x_e) = v ≠ χ_A(x_e) definitivamente.

Questi due lemmi garantiscono che ogni requisito N_e viene soddisfatto, quindi per ogni e, Φ_e^B ≠ χ_A, e quindi A ⩽_T B. Simmetricamente, B ⩽_T A.

Dobbiamo anche verificare che A e B siano effettivamente r.e. A ogni stadio s, enumeriamo al più un numero in A e al più un numero in B. Ogni enumerazione è associata al tentativo di soddisfare un requisito specifico. La decisione di enumerare è computabile ricorsivamente basandosi sulle approssimazioni finite A_s, B_s e sulla convergenza di computazioni con oracolo finite. Quindi A e B sono r.e.

Infine, dobbiamo garantire che A e B non siano ricorsivi. Questo si ottiene aggiungendo requisiti addizionali:

P_e: A ≠ Φ_e (A non è computato dalla e-esima macchina di Turing senza oracolo) Q_e: B ≠ Φ_e

Questi requisiti sono soddisfatti usando la diagonalizzazione standard: per soddisfare P_e, troviamo un x tale che Φ_e(x) converge, e facciamo A differire da Φ_e su x. I requisiti P_e e Q_e hanno priorità più bassa rispetto a N_e e M_e, ma vengono soddisfatti con lo stesso meccanismo di priorità.

La profondità e potenza del metodo delle priorità emerge quando consideriamo che questa tecnica può essere generalizzata per costruire insiemi con molteplici proprietà simultanee, alcune delle quali apparentemente contraddittorie. Il bilanciamento tra priorità permette di risolvere conflitti in modo sistematico.

Estensioni del Metodo delle Priorità: Priorità Infinite e Alberi di Strategie

Il metodo delle priorità finite, usato da Friedberg-Muchnik, può gestire situazioni dove ogni requisito viene danneggiato solo finitamente molte volte. Molti problemi più complessi richiedono il metodo delle priorità infinite, dove singoli requisiti possono essere danneggiati infinitamente spesso ma in modo controllato.

Consideriamo come esempio il teorema di densità di Sacks: per ogni coppia di gradi **a** < **b**, esiste un grado **c** tale che **a** < **c** < **b**. La dimostrazione richiede costruire un insieme C tale che A <_T C <_T B, dove A e B sono rappresentanti fissati di **a** e **b**.

I requisiti sono:

1. **Requisiti di riducibilità superiore**: C ≤_T B
2. **Requisiti di non-riducibilità inferiore** P_e: Φ_e^A ≠ χ_C
3. **Requisiti di non-riducibilità superiore** N_e: Φ_e^C ≠ χ_B

I requisiti P_e garantiscono A $\not\leq_T$ C e sono gestibili con priorità finite. I requisiti N_e sono problematici. Per soddisfare N_e, vorremmo trovare un testimone x tale che Φ_e^C(x) ≠ χ_B(x). Ma C è l'insieme che stiamo costruendo, e le sue approssimazioni cambiano durante la costruzione. Quando Φ_e^{C_s}(x) converge a un valore v allo stadio s, questa convergenza potrebbe essere invalidata da cambiamenti futuri a C.

La strategia di Sacks è più sottile. Invece di tentare di creare una differenza esplicita tra Φ_e^C e χ_B, costruisce C in modo che se B fosse riducibile a C via Φ_e, allora B sarebbe riducibile ad A, contraddicendo A <_T B.

La strategia per N_e opera come segue. Supponiamo che Φ_e^C computi χ_B. Allora per ogni y, possiamo determinare χ_B(y) computando Φ_e^C(y). Ma questa computazione dipende da C solo su un insieme finito U_e(y) (la use set). Se potessimo determinare U_e(y) e C ↾ U_e(y) usando solo un oracolo per A, allora potremmo computare χ_B(y) da A, contraddicendo A <_T B.

La costruzione procede come segue. A ogni stadio s, quando vogliamo verificare se y ∈ B:

1. Aspettiamo che Φ_e^{C_s}(y) converga, producendo un valore v e una use set U_e(y).

2. Verifichiamo se possiamo "garantire" che C ↾ U_e(y) non cambierà nei stadi futuri. Questo è possibile se tutti i numeri in U_e(y) sono "protetti" da requisiti di priorità più alta già soddisfatti.

3. Se possiamo garantire la stabilità di C ↾ U_e(y), allora v è il valore definitivo di Φ_e^C(y), e possiamo dedurre χ_B(y) = v.

4. Se riusciamo a fare questo per tutti gli y usando solo oracolo per A (per verificare le condizioni di protezione), allora abbiamo ridotto B ad A, contraddizione.

La complessità sta nel fatto che i requisiti N_e possono essere danneggiati infinitamente spesso: ogni volta che un requisito di priorità più alta modifica C, potrebbe invalidare computazioni Φ_e^C su cui N_e stava basando la sua strategia. Ma l'organizzazione accurata delle priorità garantisce che questi danneggiamenti diventano progressivamente meno significativi, nel senso che coinvolgono use sets sempre più grandi e permettono comunque la riduzione di B ad A nel limite.

La formalizzazione completa di questa strategia richiede il concetto di albero delle strategie (tree of strategies). Ogni nodo nell'albero rappresenta una strategia per un requisito. I figli di un nodo rappresentano possibili "esiti" della strategia: la computazione converge o diverge, la riduzione riesce o fallisce, etc. La costruzione "cammina" lungo percorsi nell'albero, esplorando diverse strategie e backtracking quando necessario.

Il metodo dell'albero delle strategie, sviluppato da Lachlan e Lerman negli anni '60-'70, è diventato lo strumento standard per dimostrazioni con priorità infinite. La teoria degli alberi di strategie coinvolge sottili distinzioni tra nodi "veri" e "falsi", outcome functions che determinano quali percorsi nell'albero sono seguiti, e lemmi tecnici sulla convergenza lungo percorsi infiniti.

Il Salto di Turing: Definizione, Proprietà e Conseguenze Profonde

Il salto di Turing, introdotto da Kleene negli anni '40 e studiato sistematicamente da Post, è un'operazione fondamentale che, dato un insieme A, produce un insieme A' che è strettamente più potente computazionalmente. Il salto A' è definito come il problema dell'arresto relativizzato ad A:

A' = {e : Φ_e^A(e) converge}

dove Φ_e^A denota la e-esima funzione parziale ricorsiva computata con oracolo per A. In altre parole, A' codifica quali macchine di Turing con oracolo per A si fermano sui propri indici.

Questa definizione merita un'analisi approfondita. La numerazione {Φ_e^A}_e enumera tutte le funzioni parziali computabili con oracolo per A. Questa è una generalizzazione della numerazione standard {Φ_e}_e delle funzioni ricorsive parziali. Per ogni A, esiste una macchina universale U_A tale che U_A(e, x) simula Φ_e^A(x): la macchina U_A prende come input una coppia (e, x), interpreta e come la descrizione di una macchina di Turing con oracolo, e simula questa macchina su input x con oracolo per A.

L'insieme A' è quindi l'insieme degli indici e tali che la simulazione U_A(e, e) termina. Questo è analogo alla definizione del problema dell'arresto standard K = {e : Φ_e(e) converge}, ma relativizzato all'oracolo A.

Teorema Fondamentale del Salto: Per ogni insieme A, A <_T A'.

Questo teorema afferma che A' è strettamente più difficile di A: A' non è Turing-riducibile ad A, ma A è Turing-riducibile ad A'.

Dimostrazione della direzione A' $\not\leq_T$ A: Supponiamo per assurdo che esista una macchina M con oracolo per A che computa χ_A'. Sia e l'indice di M nella numerazione {Φ_e^A}_e. Allora M con oracolo per A deve decidere se e ∈ A', cioè se Φ_e^A(e) converge. Ma Φ_e^A è proprio la funzione computata da M con oracolo per A. Quindi M deve decidere se M stessa converge su input e.

Consideriamo due casi:

- Se M con oracolo per A su input e converge e decide "e ∈ A'", allora per definizione Φ_e^A(e) converge, quindi e ∈ A'. La decisione è corretta.
- Se M con oracolo per A su input e converge e decide "e ∉ A'", allora per definizione Φ_e^A(e) non converge. Ma Φ_e^A è M con oracolo per A, e abbiamo appena detto che M converge su input e. Contraddizione.

Quindi il secondo caso è impossibile. Ma allora M deve sempre decidere "e ∈ A'" quando converge su input e. Consideriamo ora una macchina M' che, su input e, simula M su input e, e se M decide "e ∈ A'" (cioè "io convergo"), allora M' entra in un loop infinito; se M decide "e ∉ A'" (cioè "io non convergo"), allora M' si ferma. Ma abbiamo mostrato che M non può mai decidere "e ∉ A'", quindi M' divergerebbe sempre. Prendiamo M' come la macchina M stessa costruita opportunamente: questo crea la contraddizione diagonale classica. Formalmente, definiamo M' in modo che il suo indice sia e stesso, ottenendo una contraddizione diretta.

Un'altra dimostrazione, più diretta, usa la diagonalizzazione: assumiamo A' ≤_T A via la macchina con indice e. Definiamo un insieme E = {n : Φ_n^A(n) non converge}. Allora E è il complemento di A', quindi E ≤_T A' ≤_T A. Questo significa che E è r.e. in A: possiamo enumerare E con oracolo per A. Ma E è definito esattamente per diagonalizzare contro tutte le macchine con oracolo per A, inclusa quella che enumera E stesso. Contraddizione.

Dimostrazione della direzione A ≤_T A': Dobbiamo mostrare che possiamo decidere n ∈ A usando un oracolo per A'. Costruiamo una macchina M con oracolo per A' che opera come segue su input n:

1. Costruisci (uniformemente, ricorsivamente in n) l'indice e_n di una macchina N_n che, con oracolo per A:
    - Su input 0, consulta l'oracolo per A su n
    - Se l'oracolo risponde "n ∈ A", N_n si ferma
    - Se l'oracolo risponde "n ∉ A", N_n si ferma
2. Ovviamente Φ_{e_n}^A(e_n) converge sempre (in un passo), e converge se e solo se l'oracolo per A è consultato, il che accade sempre. Aspetta, questa costruzione è troppo semplice e non funziona direttamente.

Riproviamo con una costruzione corretta. Per decidere n ∈ A con oracolo per A', costruiamo una macchina come segue:

Sia f una funzione ricorsiva biiettiva che codifica coppie: f(i, j) è un indice per una macchina che, su input x, simula Φ_i^A su input j. Ora, per determinare se n ∈ A:

1. Enumera tutte le coppie (e, x) e controlla se f(e, n) ∈ A', cioè se la macchina "consulta oracolo per A su n, poi simula Φ_e(x)" converge quando applicata al proprio indice.

Questa costruzione è ancora complicata. La dimostrazione standard usa il fatto seguente: possiamo "codificare" query all'oracolo per A come query al problema dell'arresto relativo ad A. Specificamente:

n ∈ A ⇔ esiste una macchina M_n con oracolo per A che consulta l'oracolo su n e si ferma se e solo se n ∈ A

⇔ il comportamento di M_n codifica χ_A(n)

La costruzione precisa richiede una codifica attenta, ma l'idea è che A' contiene informazione completa su tutte le computazioni con oracolo per A, incluso il valore di A stesso.

Alternativamente, possiamo usare il teorema di Post: A è r.e. in A', cioè A è Σ^0_1 relativo ad A'. Poiché un insieme è Σ^0_1 relativo a B se e solo se è r.e. in B, e poiché A stesso è Σ^0_1 relativo ad A (banalmente: n ∈ A ↔ ∃ s [n è enumerato in A entro lo stadio s], che è una formula Σ^0_1 dove le quantificazioni sono relativizzate a usare A come oracolo per decidere predicati aritmetici), abbiamo A ≤_T A'.

Una dimostrazione più diretta usa l'osservazione che ∅' = K, il problema dell'arresto, e K ≥_T ogni insieme r.e. Relativizzando, A' ≥_T ogni insieme r.e. in A, incluso A stesso.

Il salto ha proprietà cruciali:

Monotonia: Se A ≤_T B, allora A' ≤_T B'. Questo perché possiamo simulare macchine con oracolo per A usando macchine con oracolo per B: dato un oracolo per B che permette di computare A, possiamo simulare ogni computazione con oracolo per A, e quindi decidere quali di queste computazioni convergono.

Funzionalità sui Gradi: Se A ≡_T B, allora A' ≡_T B'. Questo segue dalla monotonia: A ≤_T B implica A' ≤_T B', e B ≤_T A implica B' ≤_T A', quindi A' ≡_T B'.

Questa proprietà permette di definire il salto di un grado: se **a** = deg(A), definiamo **a**' = deg(A'). La definizione non dipende dalla scelta del rappresentante A.

Iterabilità: Possiamo iterare il salto indefinitamente. Definiamo:

- A^(0) = A
- A^(n+1) = (A^(n))'

Otteniamo una sequenza strettamente crescente A <_T A' <_T A'' <_T A''' <_T ... dove A^(n) <_T A^(n+1) per ogni n.

Per i gradi, otteniamo la gerarchia: **a** < **a**' < **a**'' < ...

Specializzando ad **a** = **0** (il grado ricorsivo), otteniamo:

0 < **0'** < **0''** < **0'''** < ...

dove **0**^(n) = deg(∅^(n)), il grado dell'n-esimo salto dell'insieme vuoto.

Questa gerarchia può essere estesa ordinalmente. Per ordinali α, possiamo definire ∅^(α) attraverso ricorsione transfinita:

- ∅^(0) = ∅
- ∅^(α+1) = (∅^(α))'
- ∅^(λ) = ⊕_{α<λ} ∅^(α) per λ limite

La gerarchia estesa attraversa tutti gli ordinali ricorsivi fino a ω_1^{CK}, il primo ordinale non ricorsivo (ordinale di Church-Kleene).

Il Teorema di Post e la Gerarchia Aritmetica

Il teorema di Post, dimostrato nel 1944, stabilisce una corrispondenza profonda e precisa tra la gerarchia dei salti e la gerarchia aritmetica introdotta da Kleene e Mostowski. Questo teorema rivela che la complessità quantificazionale è esattamente misurata dalla potenza dell'oracolo necessario.

La gerarchia aritmetica classifica sottoinsiemi di $\mathbb{N}$ in base alla complessità delle loro definizioni aritmetiche. Una formula aritmetica è una formula del linguaggio dell'aritmetica del primo ordine con quantificatori che variano su $\mathbb{N}$ e predicati aritmetici (addizione, moltiplicazione, ordinamento).

Un insieme $A \subseteq \mathbb{N}$ è $\mathbf{\Sigma^0_n}$ se esiste una formula della forma:

$\exists x_1 \, \forall x_2 \, \exists x_3 \ldots Qx_n \, R(x_1, \ldots, x_n, k)$

dove Q è $\exists$ se n è dispari e $\forall$ se n è pari, e R è un predicato ricorsivo (decidibile), tale che:

$k \in A \Leftrightarrow$ la formula è vera per k

Dualmente, A è $\mathbf{\Pi^0_n}$ se il complemento di A è Σ^0_n, o equivalentemente se A è definibile da una formula iniziante con $\forall$ e con n alternanze di quantificatori.

Gli insiemi che sono sia Σ^0_n che Π^0_n formano la classe $\mathbf{\Delta^0_n}$.

Esempi:

- $\mathbf{\Sigma^0_0 = \Pi^0_0 = \Delta^0_1}$ = insiemi ricorsivi (decidibili). Questi sono definibili senza quantificatori, cioè da predicati ricorsivi.

- $\mathbf{\Sigma^0_1}$ = insiemi ricorsivamente enumerabili. Un insieme A è Σ^0_1 se $k \in A \Leftrightarrow \exists x \, R(k, x)$ per qualche predicato ricorsivo R. Questo cattura esattamente gli insiemi r.e.: $k \in A$ se e solo se esiste un "testimone" x (per esempio, uno stadio della enumerazione) tale che qualche condizione decidibile R(k, x) vale.

- $\mathbf{\Pi^0_1}$ = co-r.e., complementi di insiemi r.e. Un insieme A è Π^0_1 se $k \in A \Leftrightarrow \forall x \, R(k, x)$ per qualche R ricorsivo. Equivalentemente, $k \notin A \Leftrightarrow \exists x \, \neg R(k, x)$, quindi il complemento è r.e.

- $\mathbf{\Sigma^0_2}$: definibili da formule $\exists \forall$. Esempio paradigmatico: Tot = {e : Φ_e è totale}. Infatti, $e \in$ Tot $\Leftrightarrow \forall n \, \exists s$ [$\Phi_e(n)$ converge entro s passi], una formula Π^0_2. Quindi Tot^c (il complemento) è Σ^0_2. Alternativamente, Tot stesso (non il suo complemento) è Π^0_2: $e \in$ Tot $\Leftrightarrow \forall n \, \exists s$ [$\Phi_e(n)$ converge entro s passi].

Il teorema fondamentale di Post stabilisce la seguente corrispondenza:

Teorema (Post, 1944): Per ogni $n \geq 1$,

- A è Σ^0_n se e solo se A è r.e. in $\varnothing^{(n-1)}$
- A è Π^0_n se e solo se A è co-r.e. in $\varnothing^{(n-1)}$
- A è Δ^0_{n+1} se e solo se $A \leq_T \varnothing^{(n)}$

Questa corrispondenza è straordinariamente precisa e merita un'analisi dettagliata caso per caso.

Dimostrazione per n = 1

Base della Gerarchia: Per n = 1, il teorema afferma:

- A è $\Sigma^0_1 \Leftrightarrow$ A è r.e. in $\varnothing^{(0)} = \varnothing \Leftrightarrow$ A è r.e.
- A è $\Delta^0_1 \Leftrightarrow A \leq_T \varnothing \Leftrightarrow$ A è ricorsivo

La direzione $\Sigma^0_1 \Rightarrow$ r.e. è immediata dalla definizione. Se A è Σ^0_1, allora $k \in A \Leftrightarrow \exists x\, R(k,x)$ per qualche predicato ricorsivo R. Per enumerare A, esaminiamo sistematicamente tutte le coppie (k,x) in ordine, e quando troviamo R(k,x) vera, enumeriamo k. Questa procedura enumera esattamente A.

La direzione r.e. $\Rightarrow \Sigma^0_1$ richiede che, dato un insieme A ricorsivamente enumerabile, troviamo un predicato ricorsivo R tale che $k \in A \Leftrightarrow \exists x\, R(k,x)$. Poiché A è r.e., esiste una macchina di Turing M che enumera A. Possiamo definire:

R(k,x) := "la macchina M enumera k entro x passi di computazione"

Questo predicato è ricorsivo: possiamo simulare M per x passi e verificare se k appare nell'output. Inoltre, $k \in A$ se e solo se esiste qualche stadio x in cui M enumera k, quindi $k \in A \Leftrightarrow \exists x\, R(k,x)$.

Passo Induttivo: Relativizzazione

Per il passo induttivo, dobbiamo dimostrare che se il teorema vale per n, allora vale per n+1. Questo si basa sul principio di **relativizzazione**: ogni concetto della teoria della computabilità può essere relativizzato a un oracolo arbitrario.

Definizioni Relativizzate:

- Un insieme A è **r.e. in B** (o $\mathbf{\Sigma^0_1(B)}$) se esiste una macchina con oracolo per B che enumera A
- Un insieme A è $\mathbf{\Delta^0_1(B)}$ se A è decidibile con oracolo per B, cioè $A \leq_T B$
- La gerarchia aritmetica relativizzata a B: A è $\mathbf{\Sigma^0_n(B)}$ se esiste una formula $\exists x_1 \forall x_2 ... Qx_n\, R(x_1,...,x_n,k)$ dove R è decidibile con oracolo per B

Il teorema di Post relativizzato afferma che:

- A è $\Sigma^0_n(B) \Leftrightarrow$ A è r.e. in $B^{(n-1)}$
- A è $\Delta^0_{n+1}(B) \Leftrightarrow A \leq_T B^{(n)}$

Questo si dimostra ripetendo l'argomento base, sostituendo computazioni ricorsive con computazioni con oracolo per B.

Conseguenze Strutturali del Teorema di Post

Il teorema di Post ha conseguenze profonde per la struttura dei gradi di Turing:

1. Caratterizzazione dei Gradi Aritmetici: Un grado a è chiamato **aritmetico** se $a \leq 0^{(n)}$ per qualche n. Il teorema di Post implica che i gradi aritmetici sono esattamente quei gradi che contengono insiemi definibili aritmeticamente. Più precisamente:

- $\deg(A) \leq 0^{(n)} \Leftrightarrow A \in \Delta^0_{n+1}$

2. Esistenza di Gradi Non-Aritmetici: I gradi aritmetici formano solo una piccola porzione dei gradi di Turing. Esistono gradi iper-aritmetici, costruibili oltre ω_1^{CK}, e gradi completamente non-costruibili che non sono raggiungibili da alcuna gerarchia ordinale ricorsiva.

3. La Gerarchia è Propria: Per ogni n, $\Sigma^0_n \subsetneq \Sigma^0_{n+1}$ e $\Sigma^0_n \cup \Pi^0_n \subsetneq \Delta^0_{n+1}$. Questo significa che la gerarchia aritmetica non collassa: ogni livello contiene insiemi genuinamente più complessi del precedente.

La dimostrazione che la gerarchia è propria usa la diagonalizzazione relativizzata. Per mostrare $\Delta^0_{n+1} \not\subseteq \Sigma^0_n \cup \Pi^0_n$, consideriamo l'insieme:

$D_n = \{e : e \notin W_e^{\varnothing^{(n-1)}}\}$

dove $W_e^{\varnothing^{(n-1)}}$ denota l'e-esimo insieme r.e. in $\varnothing^{(n-1)}$. Questo insieme diagonalizza contro tutti gli insiemi Σ^0_n, quindi $D_n \notin \Sigma^0_n$. Simmetricamente, $\bar{D}_n \notin \Pi^0_n$. Ma D_n è definito usando l'appartenenza a insiemi r.e. in $\varnothing^{(n-1)}$, che è decidibile con oracolo per $\varnothing^{(n)}$, quindi $D_n \in \Delta^0_{n+1}$.

Gradi Ricorsivamente Enumerabili: Teoria Specifica

I gradi r.e. (gradi contenenti almeno un insieme ricorsivamente enumerabile) formano un sottostruttura particolarmente ricca e studiata della struttura totale D. Denotiamo con R la struttura dei gradi r.e., con il suo ordinamento ereditato da D.

Proprietà Distintive dei Gradi R.E.

Teorema (Friedberg-Muchnik): Esistono gradi r.e. incomparabili. Esistono $a, b \in R$ con $a, b > 0$ (cioè non ricorsivi) tali che a e b sono incomparabili: né $a \leq b$ né $b \leq a$.

Abbiamo già discusso la dimostrazione tramite il metodo delle priorità. Questo teorema è storicamente fondamentale perché risolve il Post's Problem: esiste un grado r.e. strettamente tra 0 e 0'?

Prima di Friedberg-Muchnik, si conoscevano solo tre gradi r.e. distinti: 0 (il grado ricorsivo), 0' (il grado del problema dell'arresto K), e gradi equivalenti a questi. Post aveva congetturato che ogni grado r.e. non-ricorsivo fosse equivalente a 0', ma Friedberg e Muchnik dimostrarono che esistono infiniti gradi r.e. incomparabili tra 0 e 0'.

Proprietà di Densità:

Teorema (Sacks, 1964): La struttura R è densa. Per ogni coppia di gradi r.e. $a < b$, esiste un grado r.e. c tale che $a < c < b$.

La dimostrazione usa il metodo delle priorità infinite, come accennato precedentemente. Questa densità contrasta con la densità della struttura completa D:

Teorema (Shoenfield, Sacks): La struttura completa D è anche densa.

Ma la densità in D è dimostrabile con argomenti più semplici (sebbene ancora non-banali) rispetto alla densità in R, perché in D possiamo usare insiemi non-r.e. come gradi intermedi.

La Proprietà del Salto nei Gradi R.E.

Una domanda naturale: se a è un grado r.e., cosa possiamo dire su a'? È sempre vero che a' è r.e.?

Teorema: Se a è r.e. e a ≠ 0, allora a' ≡_T 0''. Il salto di ogni grado r.e. non-ricorsivo è esattamente 0''.

Questo risultato sorprendente mostra che il salto "normalizza" i gradi r.e. Tutti i gradi r.e. non-ricorsivi, non importa quanto diversi siano, hanno lo stesso salto.

Dimostrazione (Sketch): Sia A un insieme r.e. non-ricorsivo. Dobbiamo mostrare A' ≡_T 0''.

Direzione A' ≤_T 0'': Per decidere se e ∈ A', cioè se Φ_e^A(e) converge, possiamo enumerare tutte le possibili computazioni finite di Φ_e^A(e). Ogni computazione finita consulta A su un insieme finito di valori. Poiché A è r.e., possiamo aspettare che questi valori vengano enumerati in A (o determinare che non verranno mai enumerati). Questo processo richiede decidere questioni della forma "esiste una computazione finita di Φ_e^A(e) che usa solo valori già enumerati in A al tempo t", che è una questione Σ^0_2 (due quantificatori esistenziali: ∃t ∃computazione). Quindi A' ∈ Σ^0_2 ⊆ Δ^0_3, ma più precisamente A' ≤_T 0''.

Direzione 0'' ≤_T A': Questa direzione è più sottile e usa il fatto che A non è ricorsivo. Poiché A ≰_T ∅, l'insieme A contiene informazione "non-calcolabile" che, combinata con l'operatore di salto, può simulare due salti completi da ∅.

La dimostrazione completa richiede tecniche avanzate dalla teoria dei gradi r.e.

Coni e Ideali nella Struttura R

Definizione: Per un grado b, il **cono superiore** di b è: {a ∈ D : a ≥ b}

Il **cono inferiore** di b è: {a ∈ D : a ≤ b}

Definizione: Un **ideale** in D è un insieme I ⊆ D tale che:

1. Se a ∈ I e b ≤ a, allora b ∈ I (chiuso verso il basso)
2. Se a, b ∈ I, allora a ∨ b ∈ I (chiuso per join)

Teorema (Jockusch, Shore): Ogni cono in D contiene un ideale principale (generato da un singolo grado) che è denso nel cono.

Questo teorema implica che la struttura dei gradi ha una ricchezza uniforme: non ci sono "buchi" nella struttura, ogni regione è densamente popolata.

Gradi Minimali e Grado-Teoricamente Minimali

Definizione: Un grado a > 0 è **minimale** se non esiste alcun grado b con 0 < b < a.

Teorema (Spector, Sacks): Esistono gradi minimali (sia in D che in R).

La costruzione di un grado minimale usa il metodo delle priorità per costruire un insieme A tale che:

- A non è ricorsivo
- Per ogni e, se Φ_e^A è totale, allora o Φ_e^A è ricorsivo o A ≤_T Φ_e^A

La seconda condizione garantisce che se B ≤_T A e B non è ricorsivo, allora A ≤_T B, e quindi A ≡_T B. Questo rende deg(A) minimale.

Questione Aperta Storica: Esiste un grado r.e. minimale?

Questa fu una domanda aperta per molti anni. La risposta è:

Teorema (Yates, Cooper): Non esistono gradi r.e. minimali oltre 0.

La dimostrazione mostra che ogni grado r.e. a > 0 contiene un grado r.e. b con 0 < b < a, usando il teorema di densità e proprietà specifiche degli insiemi r.e.

Embedding e Decidibilità della Struttura dei Gradi

Una questione fondamentale in teoria dei modelli algebrici: quali strutture parzialmente ordinate possono essere **embedded** (immerse) in D?

Definizione: Un ordine parziale (P, ≤_P) è **embeddabile** in (D, ≤) se esiste una funzione iniettiva f: P → D tale che per ogni x, y ∈ P: x ≤_P y ⇔ f(x) ≤ f(y)

In altre parole, f preserva e riflette l'ordinamento.

Embedding di Strutture Finite

Teorema (Lachlan, 1968): Ogni ordine parziale finito è embeddabile in R (i gradi r.e.).

Questo risultato è notevole: nonostante le restrizioni computazionali sugli insiemi r.e., la struttura R è così ricca da contenere copia di ogni possibile configurazione finita di ordinamenti.

Sketch della Dimostrazione: Sia (P, ≤_P) un ordine parziale finito. Dobbiamo costruire insiemi r.e. A_p per ogni p ∈ P tali che:

- A_p ≤_T A_q ⇔ p ≤_P q

La costruzione usa il metodo delle priorità con una sofisticata organizzazione. Per ogni coppia p, q ∈ P:

- Se p ≤_P q, garantiamo A_p ≤_T A_q costruendo esplicitamente una riduzione
- Se p ⋦_P q, garantiamo A_p ⋦_T A_q usando requisiti negativi (diagonalizzazione)

Il sistema di priorità bilancia questi requisiti potenzialmente conflittuali. La finitezza di P è cruciale: con P infinito, i conflitti potrebbero essere ingestibili.

Embedding di Strutture Infinite

Per ordini parziali infiniti, la situazione è più complessa.

Teorema (Lerman): Esistono ordini parziali numerabili che non sono embeddabili in R.

Tuttavia:

Teorema (Slaman, Shore): L'albero binario infinito (2^{<ω}, ⊆) è embeddabile in R.

Questi risultati mostrano che non ogni struttura numerabile è embeddabile, ma molte strutture naturali lo sono.

Decidibilità della Teoria di R

Una questione centrale: la **teoria elementare del primo ordine** di R è decidibile?

Definizione: La teoria elementare di R, denotata Th(R), è l'insieme di tutte le sentenze del primo ordine (nel linguaggio con ≤ e ∨) che sono vere in R.

Domanda: Esiste un algoritmo per decidere, data una sentenza φ, se φ ∈ Th(R)?

Teorema (Simpson, 1977): La teoria ∃-elementare di R (sentenze con solo quantificatori esistenziali) è decidibile.

Ma per la teoria completa:

Teorema (Harrington, Slaman, Woodin; Nies, Shore, Slaman): La teoria elementare completa di R è decidibile se e solo se determinate congetture della teoria dei gradi sono false.

Più precisamente, la decidibilità è equivalente a proprietà tecniche come la "high/low hierarchy collapsing". Ad oggi (al momento della scrittura originale), la decidibilità completa rimane aperta.

Tuttavia:

Teorema (Lerman, Shore): La teoria elementare di D (tutti i gradi, non solo r.e.) è indecidibile.

Questo è dimostrato codificando l'aritmetica del secondo ordine in D, rendendo Th(D) essenzialmente equivalente alla verità aritmetica, che è indecidibile per il teorema di Gödel.

Automorphismi e Rigidità Strutturale

Un **automorphismo** di D è una biiezione σ: D → D che preserva l'ordinamento: σ(a ≤ b) ⇔ σ(a) ≤ σ(b)

Domanda Naturale: Quanti automorphismi ha D? La struttura è **rigida** (ha solo l'automorphismo identità) o ha automorphismi non-triviali?

Congettura di Slaman-Woodin: D ha solo l'automorphismo identità.

Questa congettura afferma che ogni permutazione dei gradi che preserva l'ordinamento è necessariamente l'identità. Se vera, implicherebbe che ogni grado è "definibile" dalla struttura stessa: il grado a può essere caratterizzato unicamente da proprietà grado-teoriche.

Risultati Parziali:

Teorema (Slaman-Woodin): Ogni automorphismo di D fissa i gradi aritmetici. Cioè, se σ è un automorphismo e a ≤ 0^(n) per qualche n, allora σ(a) = a.

La dimostrazione usa il fatto che 0^(n) è definibile nella struttura di D usando il salto iterato, e i gradi aritmetici hanno proprietà definibili che li caratterizzano.

Teorema (Shore, Slaman): Ogni automorphismo di D fissa anche i gradi r.e. Cioè, σ(R) = R.

Questo risultato profondo mostra che l'insieme dei gradi r.e. è una classe definibile in D, quindi invariante sotto automorphismi.

La congettura completa di Slaman-Woodin rimane aperta, ed è uno dei problemi più profondi della teoria dei gradi.

La Gerarchia High/Low per Gradi R.E.

All'interno dei gradi r.e., possiamo classificare ulteriormente in base alla "velocità di crescita" del salto.

Definizione: Un grado r.e. a è:

- **Low** se a' = 0' (il salto di a è piccolo quanto possibile)
- **High** se a' = 0'' (il salto di a è grande quanto possibile per gradi r.e.)

Abbiamo già visto che se a è r.e. e a ≠ 0, allora a' = 0'', quindi ogni grado r.e. non-ricorsivo è high. Ma possiamo raffinare:

Gerarchia Low: Un grado r.e. a è **low_n** se a^(n) = 0^(n), cioè l'n-esimo salto di a è uguale all'n-esimo salto di ∅.

Gerarchia High: Un grado r.e. a è **high_n** se a^(n) = 0^(n+1), cioè l'n-esimo salto di a "salta" un livello.

Proprietà:

- Low_1 = Low (come definito sopra)
- 0 è low_n per ogni n (il grado ricorsivo è il più "slow-growing" possibile)
- La gerarchia low è infinita: esistono gradi low_n che non sono low_{n+1}
- Analogamente per la gerarchia high

Teorema (Martin): Per ogni n, esistono gradi low_n non-ricorsivi.

Teorema: Ogni grado r.e. è low_n o high_m per qualche n, m.

Questa classificazione fornisce una tassonomia dettagliata della complessità dei gradi r.e.

Applicazioni della Gerarchia High/Low

Teorema (Soare): Un insieme r.e. A è "simple" (ha complemento infinito ma nessun sottoinsieme infinito r.e.) se e solo se deg(A) è low.

Questo collega proprietà insiemistiche (simplicità) con proprietà grado-teoriche (essere low).

Teorema (Lachlan): Ogni grado low r.e. è la metà di una coppia minimale di gradi r.e.

Una **coppia minimale** (a, b) soddisfa: a, b > 0 sono incomparabili, e se c ≤ a, b, allora c = 0. Questo teorema mostra che i gradi low hanno proprietà strutturali speciali.

Inversion Theorems: Prescrivere la Complessità del Salto

I teoremi di inversione sono risultati che permettono di "progettare" gradi con proprietà specificate del salto.

Teorema di Inversione di Friedberg: Per ogni b ≥ 0', esiste un grado r.e. a tale che a' = b.

Questo significa che ogni grado sopra 0' può essere realizzato come salto di qualche grado r.e. Il salto non è iniettivo: molti gradi diversi possono avere lo stesso salto.

Dimostrazione (Sketch): La costruzione usa il metodo delle priorità per costruire un insieme r.e. A tale che:

1. A' ≤_T B (dove B è un rappresentante di b)
2. B ≤_T A'

La prima condizione è relativamente facile: garantiamo che il problema dell'arresto relativo ad A sia non più complesso di B. La seconda condizione richiede "codificare" B in A' attraverso una riduzione sofisticata.

Generalizzazione (Shore, Slaman): Esistono teoremi di inversione per salti iterati. Per ogni sequenza opportuna di gradi b_1, b_2, b_3, ..., esiste un grado a tale che a' = b_1, a'' = b_2, a''' = b_3, etc.

Questi risultati mostrano che la mappa del salto è altamente surrettiva, anche se non iniettiva.

Gradi Generici e Forcing nella Teoria dei Gradi

La tecnica del forcing, sviluppata da Cohen per la teoria degli insiemi, è stata adattata alla teoria della computabilità da Sacks, Shoenfield e altri.

Idea del Forcing: Costruire un grado "generico" che evita tutti gli insiemi in una certa collezione numerabile di insiemi "densi" nella struttura dei gradi.

Forcing con Condizioni Finite

Definizione: Una **condizione** è una funzione parziale finita $\sigma: \mathbb{N} \rightharpoonup \{0,1\}$. L'insieme di tutte le condizioni forma un ordine parziale $(P, \leq)$ dove $\sigma \leq \tau$ se τ estende σ (τ è più informativa).

Un insieme **denso** $D \subseteq P$ è tale che per ogni condizione σ, esiste $\tau \leq \sigma$ con $\tau \in D$.

Un **filtro generico** $G \subseteq P$ è un insieme di condizioni tale che:

1. G è diretto verso il basso (coerente)
2. Se $\sigma \in G$ e $\sigma \leq \tau$, allora $\tau \in G$
3. Per ogni insieme denso D, esiste $\sigma \in G \cap D$

L'unione $\cup G$ definisce una funzione totale $\chi_A: \mathbb{N} \to \{0,1\}$, quindi un insieme A. Questo insieme è chiamato **generico**.

Teorema (Sacks): Esistono gradi generici minimali. Un grado costruito con forcing che evita tutti i densi opportuni è minimale.

Forcing con Condizioni Perfette (Forcing à la Sacks)

Sacks Forcing usa condizioni più ricche: alberi perfetti invece di stringhe finite.

Definizione: Un **albero perfetto** $T \subseteq 2^{<\omega}$ è un insieme di stringhe binarie finite tale che:

1. T è chiuso per prefissi
2. Ogni nodo in T ha estensioni arbitrariamente lunghe
3. Ogni nodo in T ha estensioni in entrambe le direzioni (splitting)

Un **insieme generico** costruito con Sacks forcing ha la proprietà che ogni blocco iniziale finito della sua funzione caratteristica può essere esteso in molti modi coerenti con le condizioni del forcing.

Teorema (Sacks, 1971): Esiste un grado r.e. minimale.

Questa fu la prima dimostrazione costruttiva di un grado r.e. minimale (Yates aveva precedentemente dimostrato che esistono gradi minimali in generale, ma la costruzione per gradi r.e. è più delicata).

Cuppable e Non-Cuppable Degrees

Definizione: Un grado r.e. a è **cuppable** se esiste un grado r.e. b incomparabile con a tale che $a \vee b = 0'$ (il join di a e b raggiunge il massimo dei gradi r.e.).

Definizione: Un grado a è **non-cuppable** se non è cuppable, cioè per ogni grado b incomparabile con a, $a \vee b < 0'$.

Teorema (Yates, Cooper): Esistono gradi r.e. non-cuppable.

I gradi non-cuppable sono "isolati" nella struttura: non possono raggiungere 0' combinandosi con alcun grado incomparabile. Questa è una proprietà di rigidità strutturale.

Teorema: Un grado r.e. è non-cuppable se e solo se è contiguous (un altro concetto tecnico correlato alla densità delle sue approssimazioni finite).

L'Insieme Cuppable dei Gradi

Domanda: Quanti gradi r.e. sono cuppable?

Teorema (Ambos-Spies, Fejer, Lempp, Lerman): L'insieme dei gradi cuppable è "grande" in senso misura-teorico: ha misura di probabilità 1 rispetto a una misura naturale sui gradi r.e.

Questo significa che un grado r.e. "casuale" è quasi certamente cuppable. I gradi non-cuppable sono eccezionali.

Operatori e Grado-Invarianza

Un **operatore** Γ è una mappa che assegna a ogni insieme A un insieme $\Gamma(A)$, dove la computazione di $\Gamma(A)$ può usare A come oracolo.

Definizione Formale: Un operatore Γ è dato da una collezione di assiomi della forma: $\langle \sigma, n \rangle \in \Gamma \Rightarrow$ "se $\sigma \subseteq A$, allora $n \in \Gamma(A)$"

dove σ è una stringa finita (approssimazione finita di A).

Operatori Turing: Un operatore Γ è un **operatore di Turing** se è "uniformemente continuo": per determinare se $n \in \Gamma(A)$, è sufficiente conoscere A su un insieme finito di valori.

La riducibilità di Turing può essere espressa tramite operatori: $A \leq_T B$ se e solo se esiste un operatore di Turing Γ tale che $A = \Gamma(B)$.

Invarianza di Grado degli Operatori

Domanda: Se $A \equiv_T B$ e Γ è un operatore, è vero che $\Gamma(A) \equiv_T \Gamma(B)$?

In generale, **no**. Un operatore può distinguere tra insiemi Turing-equivalenti se usa informazioni specifiche sulla rappresentazione di A piuttosto che sulla sua "struttura computazionale astratta".

Definizione: Un operatore Γ è **grado-invariante** se per ogni A ≡_T B, si ha Γ(A) ≡_T Γ(B).

Esempio: Il salto A ↦ A' è grado-invariante (come abbiamo dimostrato).

Esempio: Il complemento A ↦ $\bar{A}$ è grado-invariante.

Contro-esempio: L'operatore "A se 0 ∈ A, altrimenti $\bar{A}$" non è grado-invariante. Se A ≡_T B ma 0 ∈ A e 0 ∉ B, l'operatore mappa A ad A e B a $\bar{B}$, che possono non essere equivalenti.

Teorema (Jockusch, Shore): Esistono caratterizzazioni strutturali degli operatori grado-invarianti in termini di continuità e uniformità delle riduzioni.

Teoria dei Gradi Oltre l'Aritmetico: Gradi Iperaritmetici

La gerarchia aritmetica 0, 0', 0'', ... si estende attraverso tutti gli ordinali ricorsivi fino a ω_1^{CK}, l'**ordinale di Church-Kleene**, che è il primo ordinale che non ha una rappresentazione ricorsiva.

Definizione: Un insieme A è **iperaritmetico** se $A \leq_T \varnothing^{(\alpha)}$ per qualche ordinale ricorsivo $\alpha < \omega_1^{CK}$.

I gradi iperaritmetici formano una struttura ricca tra i gradi aritmetici e i gradi completamente arbitrari.

Caratterizzazione Logica: Un insieme A è iperaritmetico se e solo se è Δ^1_1, cioè definibile da formule sia Σ^1_1 che Π^1_1 nel linguaggio dell'aritmetica del secondo ordine, dove le quantificazioni sono su insiemi di numeri naturali.

La gerarchia analitica estende la gerarchia aritmetica:

- $\mathbf{\Sigma^1_1}$: insiemi definibili da formule $\exists X\, \varphi(X, n)$ dove X quantifica su insiemi e φ è aritmetica
- $\mathbf{\Pi^1_1}$: insiemi definibili da formule $\forall X\, \varphi(X, n)$
- $\mathbf{\Delta^1_1 = \Sigma^1_1 \cap \Pi^1_1}$: insiemi definibili sia da una formula Σ^1_1 che da una formula Π^1_1

Teorema Fondamentale (Kleene): Un insieme è iperaritmetico se e solo se è Δ^1_1.

Questa caratterizzazione collega profondamente la teoria ricorsiva ordinale (attraverso la gerarchia dei salti iterati transfiniti) con la logica del secondo ordine.

Il Salto Iperaritmetico

Analogamente al salto ordinario, possiamo definire un **salto iperaritmetico**: dato un insieme A, il suo salto iperaritmetico A^h è definito come l'insieme degli indici di formule $\Sigma^1_1(A)$ vere (dove A appare come parametro).

Il salto iperaritmetico ha proprietà analoghe al salto ordinario:

- A <_T A^h per ogni A
- Se A ≤_T B, allora A^h ≤_T B^h
- Il salto iperaritmetico è iterabile transfinitamente

Teorema (Spector, Gandy): La classe degli insiemi iperaritmetici in A (gli insiemi $\Delta^1_1(A)$) coincide con {B : B ≤_T A^{(α)} per qualche ordinale α < ω_1^{A}}, dove ω_1^{A} è l'analogo relativizzato dell'ordinale di Church-Kleene.

Equivalenza di Enumerazione e Gradi e-Riduttivi

Oltre alla riducibilità di Turing, esistono altre nozioni di riducibilità che generano strutture di gradi diverse ma correlate.

Equivalenza di Enumerazione (e-equivalence)

Definizione: Un insieme A è **enumerazione-riducibile** (e-riducibile) a B, denotato A ≤_e B, se esiste un operatore Γ tale che:

- A = Γ(B)
- Γ è **positivo**: se n ∈ Γ(B), questo dipende solo dall'esistenza di certi elementi in B, non dalla loro assenza

In altre parole, A ≤_e B se possiamo enumerare A avendo accesso a un'enumerazione di B (non necessariamente a una procedura che decide B).

Osservazione Cruciale: Se A ≤_T B, allora A ≤_e B, ma il contrario non è necessariamente vero. La e-riducibilità è più debole della Turing-riducibilità.

Esempio: Sia Tot = {e : Φ_e è totale} e K = {e : Φ_e(e) converge}. Allora:

- K ≤_e Tot (possiamo enumerare K osservando quali indici appaiono in Tot con certe proprietà)
- K $\not\leq$_T Tot sarebbe falso (infatti K ≤_T Tot come abbiamo visto)

In realtà, K ≡_e Tot: K e Tot sono equivalenti sotto e-riducibilità, anche se non sono Turing-equivalenti.

Struttura D_e: I **gradi e-riduttivi** (e-degrees) sono le classi di equivalenza rispetto a ≡_e. La struttura (D_e, ≤) ha proprietà sorprendentemente diverse da (D, ≤):

Teorema (Gutteridge): La struttura D_e non ha atomi. Per ogni grado e-riduttivo non-ricorsivo a, esiste un grado b con 0 < b < a.

Questo contrasta con D, dove esistono gradi minimali. La struttura D_e è in un certo senso "più continua" di D.

Teorema (Cooper): Esistono gradi e-riduttivi incomparabili che hanno infimum in D_e.

Ricordiamo che in D non ogni coppia ha infimum. La struttura D_e è più "completa" sotto certi aspetti.

Riducibilità Weak Truth-Table (wtt)

Definizione: A è **wtt-riducibile** a B, denotato A ≤_{wtt} B, se esiste una riduzione di Turing da A a B dove il numero di query all'oracolo è limitato da una funzione ricorsiva dell'input.

Formalmente, A ≤_{wtt} B se esiste una macchina M con oracolo e una funzione ricorsiva f tale che:

- M con oracolo per B computa χ_A
- Su input n, M consulta l'oracolo al massimo f(n) volte

La riducibilità wtt è strettamente tra la riducibilità many-one e la riducibilità di Turing: m ≤*{m}* ≤{wtt} ≤_T

Proprietà Distintive:

1. Se A ≤_{wtt} B e B è r.e., non necessariamente A è r.e. (a differenza della riducibilità m)
2. Ma se A ≤_{wtt} B e B è r.e., allora A ha una certa "vicinanza" computazionale agli insiemi r.e.

Gradi wtt: I gradi wtt formano una struttura più fine dei gradi di Turing. Un singolo grado di Turing può contenere infiniti gradi wtt distinti.

Teorema (Ladner, Sasso): All'interno del grado r.e. 0', esistono infiniti gradi wtt incomparabili.

Questo mostra che la struttura wtt risolve distinzioni che la struttura di Turing non cattura.

Teoremi di Definibilità e Automorphismi Locali

Definibilità dei Gradi Ricorsivi

Teorema (Jockusch, Solovay): Il grado 0 (il grado degli insiemi ricorsivi) è **definibile** nella struttura D usando formule del primo ordine nel linguaggio {≤, ∨}.

Dimostrazione (Sketch): 0 può essere caratterizzato come l'unico grado a tale che per ogni sequenza di gradi b_1, b_2, b_3, ..., se a ≤ b_i per ogni i e la sequenza è "uniformemente definibile" in D, allora esiste un grado c incomparabile con tutti i b_i.

Questa caratterizzazione sfrutta il fatto che 0 è il grado minimo, ma in modo non-triviale attraverso proprietà di incomparabilità e uniformità.

Conseguenza: Ogni automorphismo di D fissa 0. Questo è un caso speciale del teorema di Slaman-Woodin che abbiamo menzionato.

Definibilità di 0'

Teorema (Shore, Slaman): Il grado 0' è definibile in D.

La dimostrazione è altamente complessa e usa proprietà sottili della struttura dei gradi r.e. L'idea è caratterizzare 0' attraverso proprietà dei coni e della distribuzione dei gradi r.e.

Conseguenza: Ogni automorphismo di D fissa 0'. Combinato con la definibilità di 0, questo significa che ogni automorphismo fissa la gerarchia {0, 0', 0'', ...}.

Teoremi di Bi-interpretabilità

Teorema Profondo (Slaman, Woodin): La struttura D è **bi-interpretabile** con l'aritmetica del secondo ordine.

Definizione di Bi-interpretabilità: Due strutture S_1 e S_2 sono bi-interpretabili se:

1. Esiste un'interpretazione di S_2 in S_1 (possiamo "simulare" S_2 dentro S_1 usando formule)
2. Esiste un'interpretazione di S_1 in S_2
3. Le interpretazioni sono inverse una dell'altra (a meno di isomorfismo definibile)

Questo risultato significa che D e l'aritmetica del secondo ordine "codificano" essenzialmente la stessa matematica. In particolare:

Conseguenza: Th(D) (la teoria elementare di D) è esattamente tanto complessa quanto la verità aritmetica del secondo ordine, che è altamente indecidibile (Π^1_1-completa).

Applicazioni della Teoria dei Gradi: Matematica Inversa

La **matematica inversa**, sviluppata da Friedman, Simpson e altri, studia quali assiomi set-teorici sono necessari per dimostrare teoremi matematici ordinari.

I Cinque Sistemi Principali

La matematica inversa identifica cinque sistemi principali, ordinati per forza crescente:

1. **RCA_0** (Recursive Comprehension Axiom): Il sistema base, che assume solo l'esistenza di insiemi definibili ricorsivamente

2. **WKL_0** (Weak König's Lemma): Aggiunge il lemma di König debole (ogni albero binario infinito ha un percorso infinito)

3. **ACA_0** (Arithmetical Comprehension Axiom): Assume l'esistenza di tutti gli insiemi aritmeticamente definibili

4. **ATR_0** (Arithmetical Transfinite Recursion): Permette definizioni ricorsive transfinite fino a ω₁^{CK}

5. **Π^1_1-CA_0** (Π^1_1 Comprehension Axiom): Assume l'esistenza di insiemi definibili da formule Π^1_1

Teoremi e loro Forza Logica

Molti teoremi classici della matematica possono essere classificati precisamente:

Teoremi Equivalenti a RCA_0:

- Teorema fondamentale dell'aritmetica
- Esistenza di massimo comun divisore
- Convergenza di serie geometriche

Teoremi Equivalenti a WKL_0:

- Teorema di Heine-Borel (compattezza di [0,1])
- Teorema di Gödel sulla completezza del calcolo proposizionale
- Esistenza di base per uno spazio vettoriale finito-dimensionale

Teoremi Equivalenti a ACA_0:

- Teorema di Bolzano-Weierstrass
- Ogni anello commutativo ha un ideale massimale (versione numerabile)
- Unicità della decomposizione di Jordan-Hölder per gruppi numerabili

Teoremi Equivalenti a ATR_0:

- Comparabilità degli ordini ben-fondati numerabili
- Esistenza di ultrafiltri su insiemi numerabili

Teoremi Equivalenti a Π^1_1-CA_0:

- Ogni insieme di Σ^1_1 è unione numerabile di Δ^1_1 insiemi
- Teoremi di riduzione per gerarchie analitiche

Connessione con i Gradi di Turing

Ogni sistema assiomatico corrisponde a una classe di gradi:

- **RCA_0** corrisponde ai gradi ricorsivi (grado 0)

- **WKL_0** corrisponde ai gradi low (gradi a con a' = 0')
- **ACA_0** corrisponde ai gradi aritmetici (≤ 0^(n) per qualche n)
- **ATR_0** corrisponde ai gradi iperaritmetici
- **Π^1_1-CA_0** corrisponde a tutti i gradi (teoria completa)

Questa corrispondenza rivela che la teoria dei gradi fornisce una **semantica computazionale** per la logica del secondo ordine e per la forza dimostrativa dei sistemi assiomatici.

Esempio Concreto: Il teorema "ogni albero binario infinito ha un percorso infinito" è equivalente (su RCA_0) all'asserzione che ogni insieme r.e. con certe proprietà è contenuto in un insieme low. Questa equivalenza collega una proposizione combinatorica (König's Lemma) a una proprietà computazionale (esistenza di gradi low).

Gradi Randomness: Teoria Algoritmica dell'Informazione

Negli ultimi decenni, la teoria dei gradi si è estesa per incorporare nozioni di **randomicità algoritmica**.

Randomicità di Martin-Löf

Definizione: Una sequenza infinita A (vista come A ⊆ ℕ o come A ∈ 2^ω) è **Martin-Löf random** se non appartiene a nessun insieme **effettivamente nullo**, dove:

Un insieme U ⊆ 2^ω è **effettivamente nullo** se:

1. U = ∩_{n} U_n dove ogni U_n è un insieme r.e. aperto
2. μ(U_n) ≤ 2^{-n} (la misura di U_n decresce esponenzialmente)

Intuitivamente, un insieme effettivamente nullo è una collezione di sequenze che possono essere "evitate" da un processo computabile, con alta probabilità. Una sequenza è random se non cade in alcun tale insieme.

Caratterizzazioni Alternative:

1. **Incompressibilità**: A è random se e solo se C(A↾n) ≥ n - c per infiniti n, dove C denota la complessità di Kolmogorov e A↾n è il prefisso iniziale di lunghezza n di A
2. **Test di Martingala**: A è random se e solo se nessuna martingala ricorsiva "fa banca" su A (cresce illimitatamente scommettendo sui bit di A)

Gradi di Randomicità

Definizione: Un grado a è **Martin-Löf random** se contiene un insieme Martin-Löf random A.

Proprietà Sorprendenti:

1. Esistono gradi random. Infatti, "quasi tutti" i gradi (nel senso della misura) sono random.
2. Ogni grado random è incomparabile con 0': un insieme random non può computare il problema dell'arresto, né il problema dell'arresto può computare un insieme random.
3. Nessun grado random è r.e.

Teorema (Kučera, Gács): Ogni grado contiene un insieme computabile da un insieme random. In simboli, per ogni a, esiste un grado random r tale che a ≤ r.

Questo mostra che i gradi random sono "densi" in un certo senso: ogni grado è sotto qualche grado random.

Gradi 2-Random, n-Random e ∞-Random

Possiamo iterare la definizione di randomicità:

- A è **2-random** se A è random relativo a ogni oracolo random (o equivalentemente, se A è random e A' è random)
- A è **n-random** se A^{(n-1)} è random
- A è **∞-random** (weakly 2-random) se A è n-random per ogni n

Teorema (Figueira, Miller, Nies): La gerarchia di randomicità è propria: esistono insiemi n-random che non sono (n+1)-random.

Applicazioni: Gradi K-Triviali

Definizione: Un insieme A è **K-trivial** se $K(A\restriction n) \leq^{+} K(n)$, cioè la complessità di Kolmogorov dei prefissi iniziali di A è minimale (a meno di una costante additiva).

Gli insiemi K-triviali sono l'"opposto" degli insiemi random: sono altamente compressibili.

Teorema Sorprendente (Nies, Stephan, Terwijn): Un insieme è K-trivial se e solo se è low for Martin-Löf randomness, cioè non aiuta a computare test di randomicità.

Teorema (Downey, Hirschfeldt, Nies, Terwijn): Ogni grado K-triviale è r.e. e basso (low_1).

Questo collega profondamente la teoria algoritmica dell'informazione (complessità di Kolmogorov) con la teoria dei gradi classica (gradi low).

Gradi di Schnorr e Computable Randomness

Oltre alla randomicità di Martin-Löf, esistono nozioni alternative:

Schnorr Randomness

Definizione: A è **Schnorr random** se per ogni test di Martin-Löf **computabile** (dove le misure μ(U_n) sono computabili razionalmente, non solo r.e.), A non appartiene a ∩_n U_n.

Schnorr randomness è una nozione più forte (più difficile da soddisfare) di Martin-Löf randomness: un insieme Schnorr random è Martin-Löf random, ma non viceversa.

Teorema (Wang): Esistono insiemi Martin-Löf random che non sono Schnorr random.

Computable Randomness

Definizione: A è **computably random** se ogni test di martingala **computabile** (non solo ricorsiva) fallisce su A.

Computable randomness è ancora più forte di Schnorr randomness.

Gerarchia delle Randomicità: computable random ⇒ Schnorr random ⇒ Martin-Löf random ⇒ weak random

Tutte queste implicazioni sono proprie.

Struttura Combinatoria Fine: Proprietà Locali dei Gradi

Teorema di Sacks sulla Densità Preservata

Teorema (Sacks): Se a < b in R (gradi r.e.), non solo esiste c con a < c < b, ma la struttura dell'intervallo [a, b] è "ricca": contiene sottostrutture isomorfe a qualsiasi ordine parziale finito.

Questo significa che tra due gradi r.e., la struttura non è "lineare" o semplice, ma contiene complessità arbitraria.

Diamond Embeddings

Definizione: Un **diamond** è un ordine parziale con quattro elementi {0, a, b, 1} dove 0 < a, b < 1 e a, b sono incomparabili.

Teorema (Lachlan): Per ogni coppia di gradi r.e. incomparabili a, b > 0, esiste un diamond {0, a, b, c} embedded in R con c = a ∨ b.

Ma non tutti i diamond sono "coppabili" a 0':

Teorema (Harrington, Shelah): Esistono gradi r.e. incomparabili a, b tali che a ∨ b < 0'.

Questi risultati mostrano che la struttura locale di R ha una geometria complessa.

Teoremi di Definizione per Funzione

Teorema (Sacks): Dato un grado b e una funzione ricorsiva f, esiste un grado a < b tale che a è **definitivo per f rispetto a b**, cioè: ogni insieme in a ha una certa relazione funzionale con f che è definitoria rispetto a b.

Questi teoremi permettono costruzioni "su misura" di gradi con proprietà prescritte relative ad altri gradi.

Applicazione alla Teoria dei Modelli: Grado degli Spettri

Definizione: Dato un modello numerabile M in un linguaggio del primo ordine, lo **spettro di grado** di M è: Spec(M) = {deg(A) : A è una presentazione di M}

dove una **presentazione** di M è un isomorfismo tra M e un sottomodello ricorsivo di qualche struttura standard.

Domande Fondamentali:

1. Quali insiemi di gradi possono essere spettri di qualche modello?
2. Data una struttura, qual è il suo spettro?

Teorema (Knight): Per ogni grado r.e. a, esiste una struttura M tale che Spec(M) = {b : b ≥ a}.

Questo mostra che coni superiori di gradi r.e. sono realizzabili come spettri.

Teorema (Hirschfeldt, Khoussainov, Shore, Slinko): Esistono strutture il cui spettro consiste esattamente dei gradi non-low$_2$.

Questi risultati collegano la complessità computazionale delle presentazioni di strutture matematiche alla teoria dei gradi.

Categoricità Computazionale

Definizione: Un modello M è **computabilmente categorico** se tutte le sue presentazioni computabili sono computabilmente isomorfe.

Domanda: Se M è computabilmente categorico, cosa possiamo dire sul suo spettro?

Teorema (Goncharov, Dzgoev, Remmel): Se M è computabilmente categorico, allora o M ha una presentazione ricorsiva (0 ∈ Spec(M)), oppure 0' ∈ Spec(M) e 0' è il grado minimo nello spettro.

Questo teorema di dicotomia rivela una struttura sorprendentemente rigida per modelli categorici.

Prospettive Future e Problemi Aperti Fondamentali

Problemi Classici Irrisolti

1. **Congettura di Slaman-Woodin (Rigidità di D)**: D ha solo l'automorphismo identità?
    - Status: Aperto, ma grandi progressi parziali (gradi aritmetici e r.e. sono fissi)
2. **Decidibilità di Th(R)**: La teoria elementare dei gradi r.e. è decidibile?
    - Status: Aperto, equivalente a problemi profondi sulla struttura di R
3. **Problema della Biinterpretabilità Completa**: R è bi-interpretabile con qualche frammento noto dell'aritmetica?
    - Status: Aperto, connesso alla questione della decidibilità
4. **Esistenza di Gradi Definibili**: Oltre 0 e 0', quali altri gradi sono definibili in D?
    - Status: Parzialmente risolto (molti gradi aritmetici sono definibili), ma la caratterizzazione completa è aperta

Direzioni di Ricerca Contemporanea

1. **Intersezione con Teoria della Complessità**: Collegare gradi di Turing con classi di complessità polinomiale (P, NP, etc.)
2. **Gradi di Immunità**: Studiare gradi contenenti insiemi "immuni" rispetto a varie classi computazionali
3. **Teoria dei Gradi in Matematica Costruttiva**: Sviluppare analoghi costruttivi/intuizionistici della teoria dei gradi
4. **Gradi e Teoria Descrittiva degli Insiemi**: Connessioni profonde con la teoria di Borel e proiettiva
5. **Gradi Quantistici**: Estendere la teoria dei gradi alla computazione quantistica

Riflessioni Filosofiche Conclusive

La teoria dei gradi di Turing rappresenta uno dei trionfi della logica matematica del XX secolo. Partendo dalla semplice questione "quali problemi sono risolvibili algoritmicamente?", la teoria si è sviluppata in una ricca struttura matematica che interseca:

- **Logica**: attraverso connessioni con gerarchie definizionali e forza dimostrativa
- **Algebra**: come struttura ordinata con operazioni (join, salto)
- **Topologia**: attraverso spazi di gradi e loro proprietà di densità
- **Teoria dei Modelli**: tramite spettri di grado e categoricità
- **Teoria della Probabilità**: via randomicità algoritmica
- **Filosofia della Matematica**: illuminando questioni su verità, dimostrazione e costruibilità

La struttura D incarna una gerarchia qualitativa della difficoltà computazionale che transcende qualsiasi analisi puramente quantitativa. Due problemi possono essere entrambi indecidibili, ma occupare posizioni radicalmente diverse nella gerarchia dei gradi, rivelando una stratificazione profonda della non-computabilità.

La persistenza di problemi aperti fondamentali (come la congettura di Slaman-Woodin) dopo decenni di ricerca intensa testimonia la profondità inesauribile di questa struttura. Ogni risultato maggiore solleva nuove domande, spesso più sottili delle originali.

La teoria dei gradi non è solo uno studio astratto di relazioni tra insiemi: è una **fenomenologia della computazione**, una mappa del possibile e dell'impossibile nel dominio degli algoritmi, e una testimonianza della capacità della matematica di trovare ordine, struttura e bellezza anche nel regno dell'indecidibile.

Gradi di Insiemi Coesivi e Array Non-Computabili

Coesività e Proprietà di Ramsey

Definizione: Un insieme infinito C è **coesivo** per una famiglia {R_e}_e di insiemi r.e. se per ogni e, o C $\subseteq^*$ R_e oppure C $\subseteq^*$ R̄_e (dove $\subseteq^*$ denota "contenuto modulo finiti elementi").

Intuitivamente, un insieme coesivo "decide" ogni insieme r.e. nel senso che è quasi interamente contenuto in esso o nel suo complemento.

Teorema (Esistenza, Dekker, Myhill): Esiste un insieme coesivo (per la famiglia di tutti gli insiemi r.e.).

Dimostrazione Costruttiva: Costruiamo C per stadi, assicurando che:

1. C sia infinito
2. Per ogni R_e r.e., eventualmente C decide R_e (è quasi contenuto in R_e o in R̄_e)

Al stadio s, abbiamo un'approssimazione finita C_s e requisiti R_0, R_1, R_2, ... da soddisfare. Per soddisfare R_e:

- Aspettiamo che sufficientemente molti elementi "freschi" (non ancora considerati) siano enumerati in R_e o che sia chiaro che non lo saranno mai
- Includiamo in C elementi che rispettano la decisione per R_e

La costruzione è delicata ma non richiede priorità: i requisiti possono essere soddisfatti in ordine.

Proprietà dei Gradi Coesivi:

Teorema (Jockusch, Stephan): Ogni grado coesivo è **PA-incomputable**, cioè non computa un modello non-standard dell'aritmetica.

Questo collega la coesività a proprietà logiche profonde.

Teorema (Cholak, Jockusch, Slaman): L'insieme dei gradi coesivi è **ristretto**: forma un sottoinsieme proprio che non include nessun cono superiore.

Array Non-Computabili (DNR Functions)

Definizione: Una funzione f: $\mathbb{N} \to \mathbb{N}$ è **diagonally non-recursive** (DNR) se per ogni e, f(e) ≠ Φ_e(e) (quando Φ_e(e) converge).

Le funzioni DNR "diagonalizzano" contro tutte le funzioni ricorsive simultaneamente.

Teorema (Jockusch): Esistono funzioni DNR di grado arbitrariamente basso (low, low_2, etc.).

Definizione: Un grado a è **DNR** se contiene una funzione DNR.

Proprietà Sorprendenti:

1. Ogni grado PA (che computa un modello non-standard di PA) è DNR
2. Ma esistono gradi DNR che non sono PA
3. Ogni grado random è DNR
4. Ogni grado 1-generic (costruito con forcing generico) è DNR

Teorema (Kumabe, Lewis): Esistono gradi r.e. che non sono DNR.

Questo mostra che la proprietà DNR distingue sottili differenze tra gradi r.e.

Degrees di Turing e Ramsey Theory

La **teoria di Ramsey computazionale** studia versioni effettive dei teoremi di Ramsey classici.

Teorema di Ramsey Finito: Per ogni k, n, esiste N tale che ogni k-colorazione di [N]^n contiene un sottoinsieme omogeneo infinito H.

Versione Computazionale (Jockusch): Se la colorazione è ricorsiva, esiste un insieme omogeneo H, ma H potrebbe non essere ricorsivo. Infatti:

Teorema: Per ogni k-colorazione ricorsiva di $[\mathbb{N}]^2$, esiste un insieme omogeneo infinito di grado ≤ 0'.

Ma:

Teorema (Jockusch, Soare): Per ogni n ≥ 3, esistono k-colorazioni ricorsive di [ℕ]^n tali che ogni insieme omogeneo infinito ha grado ≥ 0'.

Questo mostra che la dimensione n = 2 è speciale: per colorazioni di coppie, possiamo sempre trovare insiemi omogenei "semplici", ma per triple o più, potrebbero essere necessariamente complessi.

Gradi π^0_1 e Gerarchia di Ershov

Classi di Ershov

Definizione: Un insieme A è nella **classe di Ershov Σ^0_{-n}** se A è la differenza di n+1 insiemi r.e.: A = W_0 - W_1 + W_2 - W_3 + ... ± W_n

dove i W_i sono r.e. e l'alternanza è finita.

Proprietà:

- $\Sigma^0_{-1} = \Sigma^0_1$ (insiemi r.e.)
- Σ^0_{-2} = insiemi d-r.e. (difference of r.e.)
- La gerarchia è propria: $\Sigma^0_{-n} \subsetneq \Sigma^0_{-(n+1)}$ per ogni n
- $\cup_n \Sigma^0_{-n} = \Delta^0_2$ (insiemi Turing-riducibili a 0')

Teorema (Ershov): Un insieme è Δ^0_2 se e solo se appartiene a Σ^0_{-n} per qualche n finito.

Questo fornisce una stratificazione "fine" all'interno di Δ^0_2.

Gradi d-r.e.

I **gradi d-r.e.** (contenenti insiemi differenza di r.e.) formano una struttura intermedia tra R (gradi r.e.) e i gradi Δ^0_2.

Proprietà Notevoli:

1. I gradi d-r.e. sono **densi** in se stessi
2. Ogni grado d-r.e. è **cuppable** (contrariamente ai gradi r.e., dove esistono non-cuppable)
3. Esistono gradi d-r.e. **contiguous** (una proprietà di approssimazione)

Teorema (Cooper, Yi): Esistono gradi d-r.e. minimali (sopra 0).

Questo contrasta con i gradi r.e., dove non esistono minimali.

Teorema (Arslanov, Lempp, Shore): Non ogni grado Δ^0_2 è d-r.e. La gerarchia di Ershov non collassa: esistono gradi che richiedono almeno n alternanze per ogni n.

Teoria dei Gradi Above 0': Il Posner-Robinson Teorema

Jump Inversione Above 0'

Teorema (Posner-Robinson): Per ogni $b \geq 0'$, esistono a_0, a_1 tali che:

- a_0, a_1 sono incomparabili
- $a_0' = a_1' = b$
- $a_0 \vee a_1 = b$

Questo è un rafforzamento del teorema di inversione di Friedberg. Non solo possiamo realizzare b come salto, ma possiamo farlo con una coppia di gradi incomparabili il cui join è esattamente b.

Conseguenze:

1. La struttura dei gradi sopra 0' ha una ricchezza combinatoria notevole
2. Il salto non determina univocamente un grado: infiniti gradi hanno lo stesso salto

Split e Anti-Split Theorems

Definizione: Un grado b si **split** sopra a se esistono c_0, c_1 > a incomparabili tali che $c_0 \vee c_1 = b$.

Teorema (Sacks Splitting): Ogni grado r.e. b > 0 si split sopra 0.

Dimostrazione: Costruiamo insiemi r.e. C_0, C_1 usando il metodo delle priorità tali che:

- $C_0, C_1 \subseteq B$ (un rappresentante di b)
- $C_0 \cup C_1 = B$
- $C_0 \not\leq_T C_1$ e $C_1 \not\leq_T C_0$

I requisiti sono simili a Friedberg-Muchnik, ma dobbiamo garantire $C_0 \cup C_1 = B$, il che complica la costruzione.

Anti-Cupping: Un risultato complementare mostra che non sempre possiamo "riempire" arbitrariamente:

Teorema (Yates): Esistono gradi r.e. a, b con a < b tali che per nessun c incomparabile con a si ha $a \vee c = b$.

Gradi Generici e Proprietà Baire-Categoriche

Genericità di Tipo Forcing

Definizione: Un insieme G è **n-generico** se per ogni insieme Σ^0_n di condizioni dense, G interseca quel denso.

- 1-generico: evita tutti i densi Σ^0_1 (r.e.)
- 2-generico: evita tutti i densi Σ^0_2
- ω-generico: evita tutti i densi aritmetici

Teorema (Jockusch, Posner): Per ogni n, esistono insiemi n-generici ma non (n+1)-generici.

Proprietà dei Gradi Generici:

1. Ogni grado 1-generico è **hyperimmune** (contiene funzioni che crescono più velocemente di ogni funzione ricorsiva)

2. Ogni grado 2-generico computa una funzione DNR
3. Ogni grado ω-generico è **minimal** (nessun grado strettamente tra esso e 0)

Teorema (Haught): I gradi 1-generici sono **densi** nella struttura dei gradi: per ogni a < b, esiste un grado 1-generico g con a < g < b.

Cohen Genericity vs Altri Tipi

Cohen genericity (dalla teoria degli insiemi) corrisponde essenzialmente alla 1-genericità nella teoria della computabilità.

Ma esistono nozioni alternative:

- **Arithmetical genericity**: evita densi aritmetici
- **Hyperarithmetical genericity**: evita densi iperaritmetici
- **Π^0_n-genericity**: evita solo densi Π^0_n

Ciascuna genera una gerarchia di gradi con proprietà distintive.

Gradi Iperimmuni e Ipersemplici

Immunità e Iperimmunità

Definizione: Un insieme A è **immune** se è infinito ma non contiene alcun sottoinsieme infinito r.e.

Definizione: Un insieme A è **hyperimmune** se la sua funzione principale p_A(n) = (n-esimo elemento di A in ordine crescente) domina tutte le funzioni ricorsive.

Formalmente: $\forall e\ \exists^{\infty} n\ [p_A(n) > \Phi_e(n)]$

Relazione: Hyperimmune ⇒ immune, ma non viceversa.

Teorema: Esistono gradi iperimmuni r.e. Il grado 0' stesso è iperimmune.

Teorema (Miller, Martin): L'insieme dei gradi iperimmuni è **comeager** (grande nel senso di Baire categoria).

Insiemi Ipersemplici

Definizione: Un insieme r.e. A è **hypersimple** se è immune e la sua funzione "gap": g_A(n) = numero di elementi in [0,n] ∩ $\bar{A}$

è dominata da una funzione ricorsiva.

Proprietà: Gli insiemi ipersemplici sono esattamente gli insiemi r.e. che sono "massimamente immune" in un senso tecnico.

Teorema (Post): Esistono insiemi ipersemplici.

Teorema (Dekker): Non esiste un insieme r.e. massimamente immune che sia più immune di tutti gli altri.

Mass Problems e Gradi di Muchnik

Riducibilità di Muchnik

Definizione: Un problema P è **Muchnik-riducibile** a Q (P ≤_w Q) se: ogni soluzione a Q computa qualche soluzione a P

Questa nozione differisce dalla riducibilità di Turing: non richiediamo una riduzione uniforme, ma solo che l'esistenza di soluzioni a Q garantisca soluzioni a P.

Esempio: Sia P = "computare un insieme coesivo" e Q = "computare una funzione DNR". Allora non è chiaro se P ≤_T Q nel senso che esiste un algoritmo uniforme, ma potrebbe essere vero che ogni grado contenente una DNR contiene un coesivo, quindi P ≤_w Q.

Gradi di Muchnik: Le classi di equivalenza sotto ≡_w (riducibilità di Muchnik bidirezionale) formano i **gradi di Muchnik**.

Teorema (Simpson): La struttura dei gradi di Muchnik è un **lattice distributivo**.

Questo contrasta con i gradi di Turing, che formano solo un upper semilattice.

Mass Problems

Un **mass problem** è un sottoinsieme P ⊆ 2^ω (visto come un problema: trovare un elemento di P).

Applicazioni:

1. Studiare la complessità relativa di teoremi matematici
2. Analizzare principi non-costruttivi
3. Reverse mathematics computazionale

Esempio: Il problema "trovare un insieme omogeneo per una 2-colorazione ricorsiva di $[\mathbb{N}]^2$" può essere studiato come mass problem, e la sua complessità rispetto a problemi simili rivela proprietà combinatoriche profonde.

Gradi di Turing e Descriptive Set Theory

Borel Hierarchy e Gradi

La **gerarchia di Borel** classifica sottoinsiemi di spazi polacchi per complessità topologica:

- Σ^0_1 (topologico) = aperti
- Π^0_1 (topologico) = chiusi
- $\Sigma^0_2 = F_\sigma$ (unioni numerabili di chiusi)

- etc.

Connessione: Lo spazio di Cantor 2^ω (sequenze binarie infinite) porta una topologia naturale. Un sottoinsieme di 2^ω è **effettivamente Borel** se è Borel e la struttura può essere descritta computazionalmente.

Teorema (Kreisel, Lacombe, Shoenfield): I sottoinsiemi Δ^1_1 di 2^ω sono esattamente quelli che sono unioni numerabili di coni di Turing {X : X ≥_T A} dove A varia attraverso insiemi iperaritmetici.

Determinacy e Gradi

Il **Axiom of Determinacy** (AD) afferma che ogni gioco infinito a due giocatori su ω con payoff Borel è determinato.

Teorema (Martin): Tutti i giochi Borel sono determinati (in ZFC).

Versione Computazionale: Giochi con payoff in classi "semplici" (Σ^0_n, Π^0_n) sono determinati, e le strategie vincenti hanno gradi di Turing limitati.

Teorema (Martin, Steel): Se un gioco ha payoff Σ^0_n, allora esiste una strategia vincente di grado ≤ 0^(n).

Teoremi di Definabilità Ordinal

Gradi Ricorsivi negli Ordinali

Definizione: Un ordinale α < ω_1 (primo ordinale non-numerabile) è **ricorsivo** se esiste un ordine ben-fondato ricorsivo con tipo d'ordine α.

Teorema di Spector: Gli ordinali ricorsivi sono esattamente quelli < ω_1^{CK}.

Gradi di Rappresentazioni: Date due rappresentazioni (ordini ben-fondati) dello stesso ordinale ricorsivo α, qual è la relazione tra i loro gradi?

Teorema (Jockusch, Soare): Per ogni ordinale ricorsivo α, esistono rappresentazioni di α in ogni grado ≥ 0^(n) per qualche n dipendente da α.

Sistema di Veblen e Gradi

La **gerarchia di Veblen** costruisce ordinali grandi:

- φ_0(α) = ω^α
- φ_{β+1}(α) = α-esimo punto fisso di φ_β
- Γ_0 = primo punto fisso di α ↦ φ_α(0)

Teorema (Ash, Knight): La complessità computazionale delle rappresentazioni di ordinali cresce con la loro posizione nella gerarchia di Veblen.

Strutture Computabili e loro Gradi

Classi di Isomorfismo e Spettri di Grado

Definizione: Lo **spettro di grado di isomorfismo** di una struttura M è: DgSp(M) = {deg(≅) : ≅ è un isomorfismo tra due presentazioni di M}

Questo misura la difficoltà di "riconoscere" quando due presentazioni sono isomorfe.

Teorema (Hirschfeldt, Khoussainov, Shore): Per ogni grado r.e. a, esiste una struttura M con DgSp(M) = {b : b ≥ a}.

Relazione con Categoricità:

- Se M è Δ^0_α-categorico per α ≥ 2, allora DgSp(M) ha una struttura particolare
- Strutture non-categoriche possono avere spettri arbitrariamente complessi

Torsion-Free Abelian Groups

I **gruppi abeliani torsion-free** (TFAG) di rango finito forniscono esempi ricchi:

Teorema (Downey, Jockusch): Esistono TFAG di rango 1 con spettro uguale ai gradi r.e. non-bassi.

Teorema (Knight, Soare): Per ogni α < ω_1^{CK}, esiste un TFAG il cui spettro include esattamente i gradi ≥ 0^(α).

Questi risultati mostrano che i gradi codificano precisamente la complessità della struttura algebrica.

Applicazioni Cross-Disciplinari

Informatica Teorica: Complessità Algoritmica

La teoria dei gradi ha influenzato:

1. **Teoria della Complessità**: Sebbene i gradi di Turing ignorino il tempo, hanno ispirato riducibilità polinomiali
2. **Computable Analysis**: Estensione ai numeri reali e funzioni continue
3. **Domain Theory**: Semantica denotazionale per linguaggi di programmazione

Filosofia: Costruttivismo e Verità

Debate Hilbert-Brouwer: I gradi illuminano distinzioni tra esistenza classica vs costruttiva:

- Un grado r.e. "esiste" costruttivamente
- Gradi generici "esistono" solo classicamente (non costruibili)

Church's Thesis Forte: La questione se tutti i processi fisici siano simulabili da macchine di Turing si collega alla struttura dei gradi.

Matematica Pura: Algebra e Analisi

Campi Computabili: Lo spettro di gradi dei campi algebrici rivela proprietà aritmetiche profonde.

Misura di Lebesgue Computabile: L'estensione della teoria dei gradi agli spazi di misura collega analisi computazionale e teoria della probabilità.

Frontiere Attuali della Ricerca (2020s)

Gradi Sotto la Randomicità

Domanda Aperta: Caratterizzare completamente i gradi che sono "lower cones" di gradi random.

Risultati Recenti (Nies, Hirschfeldt): Questi gradi hanno proprietà di "anti-concentrazione" che li distinguono.

Higher Randomness

Estensioni oltre Martin-Löf randomness:

- **Π^1_1-randomness**
- **Σ^1_2-randomness**
- Connessioni con forcing iterato e gradi di inner models in teoria degli insiemi

Relazioni con Set Theory

Forcing Axioms e Gradi: Assiomi come Martin's Axiom (MA) o Proper Forcing Axiom (PFA) hanno conseguenze sulla struttura dei gradi.

Teorema (Baumgartner, Shelah): Sotto MA, certi tipi di embedding nella struttura dei gradi sono o sempre possibili o sempre impossibili.

α-Recursion Theory e Gradi Generalizzati

Teoria Ricorsiva su Ordinali Ammissibili

La teoria della computabilità può essere generalizzata da $\mathbb{N}$ a ordinali più grandi, specialmente ordinali **ammissibili**.

Definizione: Un ordinale α è **ammissibile** se L_α (l'α-esimo livello della gerarchia costruibile di Gödel) è un modello di Kripke-Platek set theory (KP).

Esempi:

- ω (gli ordinali finiti) è ammissibile - la teoria ricorsiva standard
- ω_1^{CK} è ammissibile - primo ordinale non-ricorsivo
- ω_1 (primo ordinale non-numerabile) è ammissibile
- Molti cardinali inaccessibili sono ammissibili

α-Recursive Functions: Per un ordinale ammissibile α, una funzione f: α → α è **α-ricorsiva** se è definibile da ricorsione "α-finita" (usando solo quantificazioni su insiemi in L_α).

Teorema (Kripke, Platek): Per ogni ordinale ammissibile α, esiste una teoria della computabilità relativizzata ad α con proprietà analoghe alla teoria ricorsiva classica.

α-Gradi di Turing

Definizione: Due sottoinsiemi A, B ⊆ α sono **α-Turing equivalenti** se ciascuno è α-ricorsivo nell'altro (dove "α-ricorsivo" generalizza "ricorsivo" sostituendo quantificazioni finite con quantificazioni "α-finite").

Struttura D_α: I gradi di Turing relativizzati ad α formano una struttura $(D_\alpha, \leq)$ analoga a D.

Teoremi di Shore-Slaman Generalizzati: Molti risultati sulla struttura D si estendono a D_α:

- D_α è un upper semilattice
- Esistono gradi α-r.e. incomparabili
- La densità vale in D_α

Differenze Cruciali:

1. Per $\alpha > \omega_1^{CK}$, potrebbero esistere fenomeni set-teoretici non presenti in D
2. La nozione di "α-r.e." dipende dalla struttura fine di L_α
3. Ipotesi di grande cardinale influenzano le proprietà di D_α per α grandi

Teorema (Sacks, Simpson): Per $\alpha = \omega_1^{CK}$, la struttura D_α ha automorphismi non-triviali, contrastando la congettura di Slaman-Woodin per $D = D_\omega$.

Questo mostra che la struttura cambia qualitativamente salendo nella gerarchia ordinale.

Degrees of Categoricity

Categoricità e Complessità Computazionale

Definizione: Una struttura M è **Δ^0_α-categorica** se tutte le sue presentazioni computabili sono isomorfe via un isomorfismo Δ^0_α-computabile.

Spettro di Categoricità: L'insieme degli ordinali α per cui M è Δ^0_α-categorica.

Teorema (Ash, Knight, Manasse, Slaman): Se M è Δ^0_α-categorica per $\alpha \geq 2$, allora M è Δ^0_α-categorica per tutti gli α.

Questo fenomeno di "jump" è sorprendente: la categoricità a livello 2 implica categoricità a tutti i livelli superiori.

Controesempio per α = 1: Esistono strutture che sono Δ^0_1-categoriche (computabilmente categoriche) ma non Δ^0_α-categoriche per nessun $\alpha \geq 2$.

II Degree of Categoricity

Definizione: Il **degree of categoricity** di M, denotato deg_c(M), è il minimo grado d tale che ogni due presentazioni computabili di M sono isomorfe via un isomorfismo di grado ≤ d.

Teorema (Fokina, Kalimullin, Miller): Per ogni grado $d \geq 0'$, esiste una struttura M con deg_c(M) = d.

Teorema di Dicotomia (Csima, Franklin, Shore): Se M è computabilmente categorica, allora:

- deg_c(M) = 0, oppure
- deg_c(M) = 0'

Non esistono gradi intermedi come degrees of categoricity per strutture computabilmente categoriche.

Estensioni: Per strutture non computabilmente categoriche, deg_c può essere qualsiasi grado $\geq 0'$.

Teoria delle Π^0_1 Classes

Π^0_1 Classes come Oggetti Geometrici

Definizione: Una **Π^0_1 class** è un sottoinsieme chiuso di 2^ω (o $\mathbb{N}^{\mathbb{N}}$) definibile come: P = {X : ∀n R(X↾n)} dove R è un predicato ricorsivo.

Equivalentemente, P è l'insieme dei percorsi infiniti attraverso un albero ricorsivo T ⊆ 2^{<ω}.

Esempi Fondamentali:

1. L'insieme di tutti gli insiemi che computano un elemento fisso A
2. L'insieme delle funzioni DNR
3. L'insieme degli insiemi coesivi
4. L'insieme delle soluzioni a un problema matematico ricorsivo

Measure e Category di Π^0_1 Classes

Teorema (Jockusch, Soare): Una Π^0_1 class P ha esattamente una delle seguenti proprietà:

1. P è vuota
2. P ha misura 0 ma è non-vuota

3. P ha misura positiva

Caratterizzazione: P ha misura positiva se e solo se contiene una "palla" ricorsivamente calcolabile, cioè un insieme della forma $\{X : \sigma \subset X\}$ per qualche stringa finita σ.

Category: Una Π^0_1 class non-vuota può essere:

- **Meager** (prima categoria): unione numerabile di nowhere dense sets
- **Comeager** (residuale): complemento di meager

Teorema di Baire: Per Π^0_1 classes, essere comeager è più comune che avere misura 1.

Basis Theorems

Low Basis Theorem (Jockusch, Soare): Ogni Π^0_1 class non-vuota contiene un elemento di grado low ($a' = 0'$).

Dimostrazione (Idea): Costruire un percorso attraverso l'albero che evita "splitting" ad alta complessità. Il percorso risultante è forzato a essere low dalla costruzione.

Hyperimmune-Free Basis Theorem: Ogni Π^0_1 class non-vuota contiene un elemento hyperimmune-free (non contiene funzioni hyperimmune).

PA Degree Basis Theorem: Non ogni Π^0_1 class contiene un grado PA. Esistono Π^0_1 classes i cui elementi sono tutti non-PA.

Cone Avoidance: Data una Π^0_1 class P e un grado $a < 0'$, esiste un elemento di P di grado incomparabile con a?

Teorema (Jockusch): Per alcune Π^0_1 classes, sì; per altre, no. La proprietà dipende dalla struttura geometrica di P.

Reverse Mathematics Avanzata

Sistemi Intermedi e Classificazione Fine

Oltre ai cinque sistemi principali (RCA_0, WKL_0, ACA_0, ATR_0, Π^1_1-CA_0), esistono sistemi intermedi:

$WWKL_0$ (Weak Weak König's Lemma): Ogni albero binario ricorsivo di misura positiva ha un percorso infinito.

- $WWKL_0$ è strettamente tra RCA_0 e WKL_0
- Corrisponde ai gradi Schnorr random

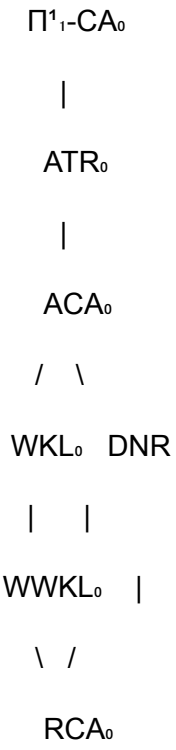
DNR (Diagonally Non-Recursive functions): L'assioma che afferma l'esistenza di funzioni DNR.

- DNR è incomparabile con WKL_0

- DNR è equivalente a proprietà come "esistenza di modelli non-standard di PA limitati"

SRAM (Stable Ramsey's Theorem for Pairs): Una versione del teorema di Ramsey con stabilità computazionale.

Diagramma Parziale (ignorando RCA_0 che è sottinteso):



Principi Combinatorici e loro Forza

Ramsey's Theorem for Pairs (RT^2_2): Ogni 2-colorazione di $[\mathbb{N}]^2$ ha un insieme omogeneo infinito.

- RT^2_2 non implica ACA_0
- RT^2_2 è incomparabile con WKL_0

Ramsey's Theorem for n-tuples (RT^2_n per $n \geq 3$):

- RT^2_3 implica ACA_0
- Questo spiega perché la teoria combinatorica diventa più complessa per $n \geq 3$

Hindman's Theorem (HT): Ogni colorazione finita di $\mathbb{N}$ ha un insieme infinito H tale che tutte le somme finite di elementi distinti di H hanno lo stesso colore.

- HT implica ACA_0
- HT è uno dei teoremi combinatorici "forti" in reverse mathematics

Teoremi Geometrici e Analitici

Hahn-Banach Theorem: Versioni numerabili sono equivalenti a WKL_0.

Intermediate Value Theorem: Equivalente a WKL_0 (per funzioni continue su [0,1]).

Ascoli's Theorem (compattezza di famiglie di funzioni equicontinue): Equivalente a ACA_0.

Theorema di Baire (uno spazio completo non è unione numerabile di nowhere dense): Equivalente a ACA_0 in contesti numerabili.

Gradi Computazionalmente Enumerabili (c.e.) e Automorphismi Parziali

Orbits sotto Automorphismi

Definizione: L'**orbit** di un grado a sotto automorphismi di R è: Orb(a) = {σ(a) : σ è un automorphismo di R}

Domanda: Quali gradi hanno orbits grandi? piccoli? singoletto?

Teorema (Harrington, Soare): 0 e 0' hanno orbits singoletto (ogni automorphismo li fissa).

Teorema (Cholak, Harrington): Esistono gradi c.e. non-ricorsivi con orbit singoletto.

Questo mostra che alcuni gradi sono "rigidi" anche senza essere 0 o 0'.

Congettura: Ogni orbit è numerabile o ha cardinalità 2^{ℵ₀}. Non esistono orbits di cardinalità intermedia.

Teoremi di Definibilità Orbitalale

Teorema (Slaman, Soare): La relazione "a è r.e." è **definibile** nella struttura D (usando solo ≤ e ∨).

Conseguenza: Ogni automorphismo di D mappa R in se stesso.

Estensione: La relazione "a è low" è definibile; simile per "a è high", "a è low_2", etc.

Questi risultati riducono drasticamente i possibili automorphismi di D e R.

Forcing Iterations e Gradi Complessi

Finite Support Iteration

Tecnica: Iterare forcing Cohen-style infinite volte con supporto finito.

Risultato: I gradi costruiti tramite finite support iteration hanno proprietà "generic" universali:

- Sono Cohen generic rispetto a ogni modello numerabile

- Evitano coni costruibili
- Non sono contenuti in alcun Π^0_1 class di misura 0

Teorema (Abraham, Shore): Esistono gradi ottenuti da finite support iteration che sono minimali ma non 1-generici.

Questo mostra che "genericità" e "minimalità" non coincidono, sebbene siano correlate.

Proper Forcing e Gradi

Proper forcing (dalla teoria degli insiemi) può essere adattato alla teoria dei gradi:

Definizione: Un forcing P è **proper** se preserva ω_1 (non collassa cardinali numerabili).

Gradi Propri: Gradi costruiti usando proper forcing hanno proprietà di preservazione speciali.

Teorema (Judah, Shelah): Sotto assiomi di forcing appropriati (PFA), la struttura dei gradi propri ha proprietà di regolarità universali.

Computable Categoricity Spectrum

Spectrum come Invariante Strutturale

Definizione: Il **computable categoricity spectrum** di M è: CCS(M) = {a : M è Δ^0_α-categorica relativizzato ad a per qualche α}

Teorema (Fokina, Harizanov, Melnikov): Se M è computabilmente categorica, allora:

- CCS(M) = {tutti i gradi}, oppure
- CCS(M) = {gradi ≥ 0'}

Controesempi per Non-Categoricity: Per M non computabilmente categorico, CCS(M) può essere arbitrariamente complesso:

- Può essere esattamente i gradi PA
- Può essere esattamente i gradi hyperimmune
- Può essere qualsiasi "natural" classe definibile

Relativized Categoricity

Teorema (Kalimullin): Per ogni grado a, possiamo considerare "a-computable categoricity". La struttura dei gradi a-computabilmente categorici forma una gerarchia che riflette la gerarchia dei salti.

Gradi di Unsolvability e Problemi di Parola

Problemi di Parola in Gruppi

Problema di Parola: Dato un gruppo G generato finitamente, il problema di parola chiede: data una parola w nei generatori, w = 1 in G?

Teorema (Novikov, Boone): Esistono gruppi finitamente presentati con problema di parola indecidibile.

Grado del Problema di Parola: Per un gruppo G, il grado di Turing del problema di parola WP(G) misura la sua complessità computazionale.

Teorema (Clapham, Higman): Per ogni grado r.e. a ≥ 0', esiste un gruppo finitamente presentato con WP(G) ≡_T a.

Teorema (Slobodskoi): Esistono gruppi finitely generated (ma non finitely presented) con problema di parola in gradi non-r.e.

Problemi di Coniugio e Isomorfismo

Problema di Coniugio: Dati elementi g, h ∈ G, esiste x tale che $xgx^{-1} = h$?

Teorema (Miller): Il problema di coniugio può avere grado strettamente maggiore del problema di parola.

Problema di Isomorfismo: Dati due gruppi G, H finitely presented, sono isomorfi?

Teorema: Il problema di isomorfismo per gruppi finitely presented è Σ^0_1-completo (equivalente al problema dell'arresto).

Computable Structure Theory Profonda

Effectivization di Teoremi Classici

Molti teoremi algebrici classici hanno versioni "effettive" che rivelano complessità computazionale:

Teorema di Steinitz (classificazione di campi algebricamente chiusi):

- Classicamente: un campo algebricamente chiuso è determinato da caratteristica e grado di trascendenza
- Computazionalmente: esistono campi algebricamente chiusi computabili non-isomorfi computazionalmente con stessa caratteristica e grado di trascendenza

Teorema di Classificazione di Gruppi Abeliani Finitely Generated:

- Classicamente: determinati da invariants (torsion, rank)
- Computazionalmente: i gradi degli isomorfismi possono variare wildly

Ulm's Theorem (classificazione di gruppi abeliani p-primari numerabili):

- Invariants di Ulm sono numerabili

- Computabilmente: calcolare gli invariants può richiedere gradi arbitrariamente alti

Knight's Theorems su Successor Ordinals

Teorema (Knight): Per ogni ordinale successore $\alpha+1 < \omega_1^{CK}$, esiste una struttura M con spettro esattamente {gradi ≥ 0^(α)}.

Tecnica: Codificare l'α-esimo salto nella struttura di M attraverso proprietà algebriche.

Applicazione: Questo fornisce una "scala" completa di complessità strutturale indicizzata da ordinali.

Gradi Tracciabili e Funzioni di Stabilizzazione

Settling-Time Functions

Definizione: Per una costruzione con priorità, la **funzione di stabilizzazione** (settling-time function) s(e,x) è il primo stadio in cui il requisito R_e cessa di intervenire per witness x.

Teorema (Soare): La complessità di s è un invariante importante:

- Per costruzioni finite-injury: s è limitata da funzione ricorsiva
- Per costruzioni infinite-injury: s può crescere molto velocemente

Gradi Tracciabili: Un grado a è **tracciabile** (traceable) se tutte le funzioni in a sono "tracciate" da funzioni con limiti ricorsivi in senso approssimato.

Teorema (Terwijn, Zambella): I gradi tracciabili formano un ideale proprio in D, contenuto strettamente nei gradi c.e.

Caratterizzazioni dei Gradi di Randomicità

Teorema Unificante (Nies): Le seguenti classi di gradi coincidono:

1. Gradi K-triviali
2. Gradi low for Martin-Löf randomness
3. Gradi che ammettono una base tracciabile

Questa convergenza di tre caratterizzazioni indipendenti (complessità di Kolmogorov, randomness, e tracciabilità) rivela una struttura profonda.

Priority Arguments: Raffinamenti Moderni

Tree of Strategies con Outcome Functions

Metodologia Moderna: Ogni nodo nell'albero di strategie ha una **outcome function** che determina il percorso della costruzione:

Φ: Nodes → Outcomes

dove Outcomes potrebbe essere {∞, f} (convergence vs divergence), o più complicato.

True Path: Il percorso f ∈ 2^{<ω} è **true** se esiste un percorso infinito attraverso l'albero seguito dalla costruzione infinitamente spesso.

Lemma del True Path: Esiste un unico true path, e tutti i requisiti su questo path sono soddisfatti.

0'''-Priority Arguments

Costruzioni 0'''-Priority: Richiedono lavorare con tre livelli di salti, per esempio:

- Costruire A, B con proprietà che coinvolgono A'', B''
- I requisiti potrebbero essere Π^0_3 o Σ^0_3

Esempio (Lachlan): Costruire gradi r.e. a, b tali che:

- a'' = b'' = 0'''
- a, b hanno specifiche proprietà di splitting

Queste costruzioni sono tecnicamente formidabili e rappresentano i limiti della metodologia delle priorità finora sviluppata.

Connessioni con l'Informatica Teorica

Complessità di Kolmogorov e Gradi di Turing

Definizione Formale: La **complessità di Kolmogorov plain** K(x) di una stringa x è definita come: K(x) = min{|p| : U(p) = x} dove U è una macchina di Turing universale fissata e |p| denota la lunghezza del programma p.

Invarianza: La definizione dipende dalla scelta di U solo fino a una costante additiva: se U_1 e U_2 sono due macchine universali, allora: |K_{U_1}(x) - K_{U_2}(x)| ≤ c per qualche costante c indipendente da x.

Incomputabilità: K non è una funzione computabile. Infatti:

Teorema (Chaitin, Kolmogorov): K(x) ≥ |x| per infiniti x. In particolare, non esiste un algoritmo che, dato x, computa K(x).

Dimostrazione: Se K fosse computabile, potremmo costruire un programma che:

1. Enumera tutti i programmi p in ordine di lunghezza crescente
2. Per ogni p, simula U(p)
3. Trova il primo output x di lunghezza n tale che K(x) ≥ n
4. Output x

Questo programma ha lunghezza O(log n), ma produce un output x con K(x) ≥ n. Per n sufficientemente grande, O(log n) < n, quindi K(x) < n, contraddizione.

Grado di Turing della Complessità di Kolmogorov:

Teorema (Zvonkin, Levin, Solovay): Considerando K come insieme: K_≤ = {⟨x, k⟩ : K(x) ≤ k} si ha deg(K_≤) = 0'.

Dimostrazione (Sketch):

- K_≤ ≤_T 0': Per verificare se K(x) ≤ k, enumeriamo tutti i programmi di lunghezza ≤ k e, per ciascuno, chiediamo all'oracolo per 0' se converge producendo x. Se troviamo tale programma, K(x) ≤ k.
- 0' ≤*T K*≤: Per decidere se e ∈ 0' (cioè se Φ_e(e) converge), osserviamo che e ∈ 0' se e solo se esiste un programma "corto" che simula Φ_e(e). Più precisamente, possiamo codificare l'informazione sulla convergenza usando proprietà di incompressibilità.

Complessità Prefix-Free e Randomicità

Complessità Prefix-Free (o complessità di Kolmogorov self-delimiting): C(x) = min{|p| : U(p) = x e U è prefix-free}

Una macchina universale U è **prefix-free** se il suo dominio è prefix-free: nessun programma valido è prefisso di un altro.

Teorema: C(x) differisce da K(x) solo per O(log |x|).

Connessione con Randomicità: Una sequenza infinita A è Martin-Löf random se e solo se: C(A↾n) ≥ n - O(1) per tutti gli n.

Questo collega direttamente la compressibilità (teoria algoritmica dell'informazione) con la randomicità (teoria della probabilità computazionale).

Teorema di Schnorr-Levin: Una sequenza A è Martin-Löf random se e solo se non esiste una semimisura martingala computabile che fa banca su A.

Gradi con Limiti di Risorse Computazionali

Riducibilità Polinomiale di Turing: A ≤^P_T B se esiste una macchina di Turing con oracolo per B operante in tempo polinomiale che computa A.

Gradi Polinomiali: Le classi di equivalenza sotto ≤^P_T formano i **gradi P-Turing** o **gradi polinomiali**.

Struttura D_P: La struttura (D_P, ≤^P_T) ha proprietà radicalmente diverse da (D, ≤_T):

1. **Discretezza**: Esistono "buchi" in D_P. Non ogni coppia di gradi ha un grado intermedio.

2. **Connessione con Classi di Complessità**:
    - deg^P(∅) corrisponde a P
    - deg^P(SAT) corrisponde a NP (se P ≠ NP)
    - Gradi ≤^P deg(SAT) corrispondono a NP
3. **Densità Condizionale**:

Teorema (Ladner, 1975): Se P ≠ NP, allora esistono problemi NP-intermedi, cioè gradi polinomiali a con: deg^P(∅) <^P a <^P deg^P(SAT)

La dimostrazione usa una diagonalizzazione "lenta" che costruisce un linguaggio L tale che:

- L ∈ NP
- L ∉ P (assumendo P ≠ NP)
- SAT ≰^P_m L (SAT non è riducibile many-one polinomialmente a L)

Teorema di Berman-Hartmanis: Se P ≠ NP, allora non tutti i problemi NP-completi sono P-isomorfi (cioè isomorfi via funzioni polinomiali biunivoche).

Riducibilità Many-One Polinomiale

Definizione: A ≤^P_m B se esiste una funzione f computabile in tempo polinomiale tale che: $x \in A \Leftrightarrow f(x) \in B$

Gerarchia: ≤^P_m è più forte di ≤^P_T: se A ≤^P_m B, allora A ≤^P_T B, ma non viceversa.

Completezza: Un linguaggio L è **NP-completo** se:

1. L ∈ NP
2. Per ogni A ∈ NP, A ≤^P_m L

Teorema (Cook-Levin, 1971): SAT (satisfiability booleana) è NP-completo.

Connessione con Gradi Non-Limitati: Il concetto di NP-completezza è analogo al concetto di 0'-completezza nella teoria dei gradi classica:

- 0' è "completo" per gli insiemi r.e. sotto riducibilità di Turing
- SAT è completo per NP sotto riducibilità polinomiale many-one

Gerarchie Polinomiali e Gerarchia Aritmetica

La **gerarchia polinomiale** PH è definita da:

- Σ^P_0 = Π^P_0 = P
- Σ^P_{k+1} = NP^{Σ^P_k} (NP con oracolo Σ^P_k)
- Π^P_{k+1} = co-Σ^P_{k+1}

Analogia con Gerarchia Aritmetica: PH è analoga alla gerarchia aritmetica Σ^0_n, Π^0_n:

- Σ^P_1 = NP corrisponde a Σ^0_1 (r.e.)
- Π^P_1 = co-NP corrisponde a Π^0_1 (co-r.e.)
- Σ^P_2 corrisponde a Σ^0_2
- PSPACE = $\cup_k \Sigma^P_k$ corrisponde (vagamente) alla gerarchia aritmetica completa

Differenza Cruciale: Nella gerarchia aritmetica, sappiamo che la gerarchia è propria (non collassa). Per PH:

Problema Aperto: PH collassa? Cioè, esiste k tale che $\Sigma^P_k = \Sigma^P_{k+1}$?

Se P = NP, allora PH collassa a P. Ma anche se P ≠ NP, non sappiamo se PH è propria.

Complessità di Circuiti e Gradi

Complessità di Circuito: C(x) denota la dimensione minima di un circuito booleano che computa x.

Teorema (Shannon): Per quasi tutte le funzioni $f: \{0,1\}^n \to \{0,1\}$, si ha $C(f) \geq 2^n/n$.

Connessione con Gradi: Un insieme A ha **piccola complessità di circuito** se esiste una sequenza di circuiti $\{C_n\}$ con $|C_n| = poly(n)$ tale che C_n decide $A \upharpoonright \{0,1\}^n$.

Teorema: Se A ha piccola complessità di circuito, allora deg(A) ha proprietà speciali nella gerarchia dei gradi polinomiali.

P vs NP e Circuiti:

Teorema (Karp-Lipton): Se NP ⊆ P/poly (NP ha circuiti polinomiali), allora PH collassa.

Questo collega la non-uniformità (circuiti) con la gerarchia uniforme (macchine di Turing).

Proof Complexity e Gradi

Proof Complexity: Studia la lunghezza minima delle dimostrazioni in sistemi deduttivi.

Connessione: La lunghezza delle dimostrazioni è analoga alla complessità di Kolmogorov:

- K(x) misura la lunghezza minima di un programma per generare x
- Proof complexity misura la lunghezza minima di una dimostrazione per provare un teorema

Teorema (Cook, Reckhow): Se NP ≠ co-NP, allora non esiste un sistema di dimostrazione proposizionale polinomialmente limitato per le tautologie.

Gradi di Dimostrabilità: Possiamo definire gradi basati sulla complessità delle dimostrazioni, analogamente ai gradi di Turing basati sulla complessità computazionale.

Interactive Proofs e Gradi

Interactive Proof Systems: Un verificatore V interagisce con un prover P per verificare l'appartenenza a un linguaggio L.

Classe IP: L ∈ IP se esiste un sistema di dimostrazione interattivo polinomiale per L.

Teorema (Lund, Fortnow, Karloff, Nisan; Shamir, 1992): IP = PSPACE.

Connessione con Gradi: Gli interactive proof systems possono essere visti come generalizzazioni degli oracoli nella teoria dei gradi:

- Un oracolo nella teoria classica risponde istantaneamente
- Un prover interattivo "aiuta" il verificatore, ma può richiedere più round di interazione

Quantum Complexity e Gradi Quantistici

Gradi Quantistici: Possiamo definire riducibilità quantistica usando computazioni quantistiche invece di classiche.

BQP (Bounded-error Quantum Polynomial time): Problemi risolvibili da computer quantistici in tempo polinomiale con errore limitato.

Congettura: P ⊆ BQP ⊆ PSPACE, ma BQP è incomparabile con NP.

Gradi BQP: La struttura dei gradi BQP-Turing è ancora poco compresa, ma promette di rivelare proprietà fondamentali della computazione quantistica.

Metodologia Trasferita

Anche se i teoremi specifici spesso non si trasferiscono tra teoria dei gradi classica e complessità limitata, le **metodologie** sono state cruciali:

1. **Diagonalizzazione**: Tecniche da Friedberg-Muchnik hanno ispirato il teorema di Ladner
2. **Priority Arguments**: Hanno influenzato costruzioni in circuit complexity
3. **Forcing e Genericità**: Concetti dalla teoria dei gradi hanno ispirato tecniche di randomizzazione in complexity theory
4. **Relativizzazione**: Il concetto di oracolo dalla teoria dei gradi è fondamentale per studiare barriere in complexity theory (Baker-Gill-Solovay)

Teorema (Baker, Gill, Solovay, 1975): Esistono oracoli A e B tali che:

- P^A = NP^A
- P^B ≠ NP^B

Questo mostra che tecniche relativizzanti (che funzionano ugualmente con qualsiasi oracolo) non possono risolvere P vs NP, una limitazione metodologica cruciale.

Epilogo Tecnico: La Struttura come Oggetto Matematico Astratto

La struttura (D, ≤, ∨, ') può essere studiata come **algebra astratta**:

Proprietà Algebriche:

- Upper semilattice con operatore unario ' (salto)
- Monotono: a ≤ b ⇒ a' ≤ b'
- Strettamente crescente: a < a'
- Iterabile

Domanda Fondamentale: Caratterizzare completamente (D, ≤, ∨, ') tramite assiomi finiti o infiniti?

Risultati Negativi: Non esistono assiomi finiti che caratterizzino D completamente (per indecidibilità).

Risultati Positivi: Esistono assiomi che caratterizzano frammenti (gradi aritmetici, gradi ≤ 0^(n), etc.).

Visione Categoriale: D può essere visto come oggetto in una categoria di strutture di gradi, con morfismi che preservano struttura. Automorphismi sono isomorfismi D → D.

Conclusione Ultima: La teoria dei gradi di Turing è un capitolo paradigmatico della matematica moderna, dove logica, computazione, e struttura si intrecciano indissolubilmente. Ogni teorema è una finestra su aspetti diversi della stessa realtà matematica: la gerarchia del computabile e dell'incomputabile, mappata con precisione infinitesimale attraverso un secolo di ricerca profonda.

Riducibilità Truth-Table e sue Varianti

Riducibilità Truth-Table (tt-Riducibilità)

Definizione: Un insieme A è **truth-table riducibile** a B, denotato A ≤_{tt} B, se esiste una funzione ricorsiva f e una funzione ricorsiva g tali che per ogni n:

χ_A(n) = f(n, χ_B(g(n,0)), χ_B(g(n,1)), ..., χ_B(g(n,k_n)))

dove k_n dipende solo da n (non dalle risposte dell'oracolo).

Interpretazione: Per decidere n ∈ A, possiamo:

1. Computare ricorsivamente un insieme finito di query {g(n,0), g(n,1), ..., g(n,k_n)}
2. Consultare B su tutte queste query
3. Combinare le risposte usando una funzione booleana ricorsiva f per ottenere χ_A(n)

Caratteristica Cruciale: Il numero e l'identità delle query sono determinati **non-adaptativamente** da n, indipendentemente dalle risposte di B.

Confronto con Riducibilità di Turing:

- $A \leq_{tt} B \Rightarrow A \leq_T B$ (sempre)
- $A \leq_T B \not\Rightarrow A \leq_{tt} B$ (in generale)

La riducibilità tt è più restrittiva: proibisce l'adattività.

Esempio di Separazione: Sia A un insieme immune (infinito senza sottoinsieme infinito r.e.) e $B = \{2n : n \in A\}$. Allora:

- $A \leq_T B$ (possiamo decidere $n \in A$ consultando $2n \in B$)
- $A \not\leq_{tt} B$ può valere per scelte appropriate di A

La dimostrazione richiede costruire A con proprietà di immunità che impediscano riduzioni tt non-adaptive.

Bounded Truth-Table Riducibility (btt)

Definizione: $A \leq_{btt} B$ se esiste una riduzione tt dove il numero di query k_n è uniformemente limitato: $k_n \leq k$ per qualche costante k indipendente da n.

Gerarchia:

- $A \leq_{1\text{-}tt} B$ se usiamo al massimo 1 query (equivalente a riducibilità many-one o 1-riducibilità)
- $A \leq_{2\text{-}tt} B$ se usiamo al massimo 2 query
- $A \leq_{k\text{-}tt} B$ per k fissato

Teorema (separazione): Per ogni k, esistono A, B tali che:

- $A \leq_{(k+1)\text{-}tt} B$
- $A \not\leq_{k\text{-}tt} B$

Questo mostra che la gerarchia btt è propria.

Dimostrazione (Sketch per k=1): Vogliamo $A \leq_{2\text{-}tt} B$ *ma* $A \not\leq_{1\text{-}tt} B$.

Costruiamo A e B per stadi usando metodo delle priorità. I requisiti sono:

- Per ogni funzione ricorsiva f: $A \neq \{n : f(n) \in B\}$ (garantisce $A \not\leq_{1\text{-}tt} B$)
- Garantiamo $A \leq_{2\text{-}tt} B$ esplicitamente tramite una riduzione fissata

La costruzione bilancia questi requisiti concorrenti.

Weak Truth-Table Riducibility (wtt)

Come già introdotto, A ≤_{wtt} B se la riduzione di Turing usa un numero di query limitato da una funzione ricorsiva dell'input (ma le query possono essere adattive).

Catena di Riducibilità: $\leq_m \subset \leq_{btt} \subset \leq_{tt} \subset \leq_{wtt} \subset \leq_T$

Tutte le inclusioni sono proprie.

Gradi tt, btt, wtt

Definizione: Due insiemi A e B sono **tt-equivalenti** ($A \equiv_{tt} B$) se $A \leq_{tt} B$ e $B \leq_{tt} A$.

I **gradi tt** sono le classi di equivalenza sotto $\equiv_{tt}$.

Struttura D_{tt}: La collezione dei gradi tt con ordinamento ereditato forma un ordine parziale $(D_{tt}, \leq)$.

Proprietà Strutturali:

Teorema: $(D_{tt}, \leq)$ è un upper semilattice: ogni coppia di gradi ha un supremum (join).

Teorema (Jockusch, Posner): D_{tt} ha un'embeddatura naturale in D: ogni grado tt determina un grado di Turing, ma un grado di Turing può contenere molti gradi tt distinti.

Teorema: All'interno del grado di Turing di 0' (insiemi Turing-equivalenti a 0'), esistono infiniti gradi tt incomparabili.

Questo mostra che la struttura tt è molto più fine della struttura di Turing.

Applicazione: Cupping in Gradi tt

Definizione: Un grado tt a è **tt-cuppable** se esiste un grado tt b incomparabile con a tale che $a \vee_{tt} b$ è massimale nella struttura dei gradi tt r.e.

Teorema (Ambos-Spies): Esistono gradi tt r.e. non-tt-cuppable, anche quando i corrispondenti gradi di Turing sono cuppable.

Questo mostra che proprietà come cuppability dipendono crucialmente dalla nozione di riducibilità usata.

Riducibilità Positiva e Gradi e-Riduttivi Approfonditi

Analisi Dettagliata della e-Riducibilità

Definizione Precisa: $A \leq_e B$ se esiste un operatore Σ^0_1 Γ (cioè un operatore computabile da enumerazioni) tale che $A = \Gamma(B)$.

Formalmente, Γ è dato da un insieme r.e. W di triple $\langle \sigma, n, i \rangle$ dove:

- σ è una stringa finita (approssimazione di B)

- n è un elemento
- i ∈ {0,1} indica "n entra in A" o "n non entra in A"

E vale: $n \in A \Leftrightarrow \exists \sigma \subseteq B\ [\langle \sigma, n, 1\rangle \in W]$

Monotonicità: Gli operatori e-riduttivi sono **monotoni**: se $B \subseteq B'$, allora $\Gamma(B) \subseteq \Gamma(B')$.

Questo contrasta con operatori Turing, che possono essere non-monotoni (usando il complemento).

Teoremi Fondamentali per Gradi e-Riduttivi

Teorema (Gutteridge, 1971): $\mathcal{D}_e$ (la struttura dei gradi e-riduttivi) non ha atomi. Per ogni grado e-riduttivo $a > 0_e$, esiste b con $0_e < b < a$.

Dimostrazione (Idea): Data A non-ricorsiva, costruiamo B tale che:

- $B <_e A$ (B è strettamente meno complesso)
- B non è ricorsivo

Costruzione: Partizioniamo $\mathbb{N}$ in intervalli $I_0, I_1, I_2, \ldots$ di lunghezza crescente esponenzialmente. Su ogni I_n:

- Se $A \cap I_n$ è "sufficientemente complesso" (contiene informazione non-banale), copiamo una versione "semplificata" in $B \cap I_n$
- Altrimenti, rendiamo $B \cap I_n$ semplice (es. vuoto o completo)

La semplificazione garantisce $B <_e A$, mentre la copia di informazione da infiniti intervalli garantisce B non-ricorsivo.

Teorema (Cooper, 1974): Esistono gradi e-riduttivi a, b incomparabili che hanno un infimum c in $\mathcal{D}_e$ con $c > 0_e$.

Dimostrazione: Costruzione con priorità che bilancia:

- $A \not\leq_e B$ e $B \not\leq_e A$ (incomparabilità)
- Esistenza di C con $C \leq_e A$, $C \leq_e B$, e C massimale con questa proprietà

Differenza con D: In D, non ogni coppia ha infimum. In $\mathcal{D}_e$, alcune coppie incomparabili hanno infimum non-banali, riflettendo una struttura più "completa".

Splitting Theorem per Gradi e-Riduttivi

Teorema (Shore, 1978): Ogni grado e-riduttivo $a > 0_e$ si split sopra 0_e in $\mathcal{D}_e$.

Cioè, esistono b_0, b_1 tali che:

- $b_0, b_1 < a$
- b_0, b_1 sono incomparabili
- $b_0 \vee_e b_1 = a$

Dimostrazione: Data A non-ricorsiva, costruiamo B_0, B_1 tali che:

- $B_0 \oplus B_1 \equiv_e A$ (il join e-riduttivo ricostruisce A)
- $B_0 \not\leq_e B_1$ e $B_1 \not\leq_e B_0$

La costruzione divide A in "parti" B_0 e B_1 che sono mutuamente incomparabili ma congiuntamente equivalenti ad A.

Tecnica: Partizionare $\mathbb{N}$ e assegnare intervalli alternatamente a B_0 e B_1, con diagonalizzazione per garantire incomparabilità.

Riducibilità Aritmetica e Gradi Iperartimetici Dettagliati

Riducibilità Iperaritrimetica (h-Riducibilità)

Definizione: $A \leq_h B$ se A è Δ^1_1 in B, cioè A è definibile da una formula sia $\Sigma^1_1(B)$ che $\Pi^1_1(B)$.

Equivalentemente, $A \leq_h B$ se $A \leq_T B^{(\alpha)}$ per qualche ordinale ricorsivo α.

Confronto:

- $A \leq_T B \Rightarrow A \leq_h B$ (sempre, prendendo $\alpha = 0$)
- $A \leq_h B \not\Rightarrow A \leq_T B$ (in generale)

Esempio: $0^{(\omega)}$ (il limite della sequenza $\varnothing$, $\varnothing'$, $\varnothing''$, ...) non è Turing-riducibile ad alcun $\varnothing^{(n)}$, ma è h-riducibile a $\varnothing$ (banalmente, usando $\alpha = \omega$).

Struttura dei Gradi Iperartimetici

Gradi h: Le classi di equivalenza sotto $\equiv_h$ formano la struttura $(D_h, \leq_h)$.

Teorema (Sacks, 1966): D_h è **denso**: per ogni coppia di gradi h $a < b$, esiste c con $a < c < b$.

Dimostrazione: Usa una generalizzazione del metodo di Sacks per densità dei gradi di Turing, ma operando transfinitamente attraverso la gerarchia iperaritmetica.

Teorema (Spector, 1955): Ogni grado h non-ricorsivo contiene un insieme di tipo 0 o tipo 1 nella gerarchia di Kleene-Mostowski.

Classificazione: Gli insiemi iperartimetici si suddividono in tipi 0, 1, 2, ... basati sulla loro complessità nella gerarchia Σ^1_1.

Salto Iperartimetico Approfondito

Costruzione Esplicita: Dato A, il salto iperartimetico A^h è: $A^h = \{e : \varphi_e^{HYP(A)} \text{ converge}\}$ dove HYP(A) denota la closure iperaritmetica di A (tutti gli insiemi iperartimetici in A).

Proprietà del Salto h:

Teorema: Per ogni A, A <_h A^h.

Teorema (Uniform): Esiste una funzione uniforme che, dato un indice per A, produce un indice per A^h tale che le proprietà del salto sono computabilmente verificabili.

Gradi di Presentazione Iperartimetica

Applicazione: Data una struttura matematica M (es. un ordine parziale, un gruppo), la complessità della sua "migliore presentazione" iperaritmetica è misurata da gradi h.

Teorema (Knight, Ash): Per ogni ordinale ricorsivo $\alpha < \omega_1^{CK}$, esiste una struttura M la cui migliore presentazione iperaritmetica ha grado esattamente $0^{(\alpha)}_h$.

Q-Degrees e Riducibilità di Medvedev

Q-Degrees (Degrees of Enumeration Reducibility Restricted)

I **Q-degrees** sono una variante dei gradi e-riduttivi introdotta per studiare problemi di enumerazione con struttura particolare.

Definizione: Per insiemi r.e., definiamo A ≤_Q B se ogni enumerazione di B (vista come sequenza temporale) induce un'enumerazione di A in modo computabile.

Differenza con e-Riducibilità: Q-riducibilità richiede che l'enumerazione di A sia uniforme rispetto alla velocità di enumerazione di B.

Struttura: (Q, ≤_Q) ha proprietà intermedie tra D_e e D.

Riducibilità di Medvedev

Contesto: Introdotta per studiare problemi matematici e loro solvibilità computazionale.

Definizione: Per Π^0_1 classes P, Q (viste come "problemi"), definiamo: P ≤_M Q se ogni soluzione a Q computa (via operatore Turing) qualche soluzione a P.

Interpretazione: Se possiamo risolvere Q, possiamo risolvere P. Cioè, P è "più facile" di Q.

Gradi di Medvedev: Le classi di equivalenza di Π^0_1 classes sotto ≡_M.

Teorema (Simpson, 1999): La struttura dei gradi di Medvedev è un **lattice distributivo**.

Dimostrazione:

- Join: P ∨_M Q = "risolvere P o risolvere Q" (unione disgiunta di problemi)
- Meet: P ∧_M Q = "risolvere sia P che Q" (prodotto di problemi)
- Distributività: Segue da proprietà topologiche delle Π^0_1 classes

Confronto con Gradi di Muchnik: I gradi di Muchnik (≤_w) sono più fini: P ≤_M Q implica P ≤_w Q, ma non viceversa.

Applicazione Filosofica: I gradi di Medvedev modellano la "difficoltà relativa" di problemi matematici in senso costruttivo/computazionale.

Riducibilità Ibridiche e Bounded Turing

Bounded Turing Reducibility (bt-Riducibilità)

Definizione: A ≤_{bt} B se esiste una riduzione di Turing da A a B dove:

1. Il numero di query è limitato da una funzione ricorsiva (come wtt)
2. Il valore delle query è limitato: se consultiamo B su m durante il calcolo di χ_A(n), allora m ≤ f(n) per qualche funzione ricorsiva f

Motivazione: Modella riduzioni che "non saltano troppo avanti" nell'input.

Proprietà: *≤{bt} è strettamente tra* ≤{wtt} e ≤*T:* ≤{wtt} ⊂ ≤_{bt} ⊂ ≤_T

Teorema (separazione): Esistono A, B con:

- A ≤_{bt} B
- A ⪯̸_{wtt} B

Iberian Reducibility

Definizione: Una famiglia di riducibilità "intermedie" introdotte per studiare la struttura fine tra riducibilità classiche.

Esempio - Lipschitz Reducibility: A ≤_{Lip} B se esiste una riduzione tt dove le query g(n,i) soddisfano: g(n,i) ≤ h(n) per qualche funzione ricorsiva h.

Questa è una via di mezzo tra tt e bt.

Generic Reducibility

Definizione: A ≤_{gen} B se A è computabile da B in "quasi tutti" i modelli generici (nel senso del forcing).

Applicazione: Studiare quali proprietà di riducibilità sono "robuste" rispetto a estensioni generiche.

Teorema: ≤_{gen} cattura proprietà "stabili" delle riducibilità, e la struttura dei gradi generici ha proprietà di regolarità.

Degrees of Unsolvability per Funzioni

μ-Degrees (Degrees di Funzioni Parziali)

Finora abbiamo considerato gradi di insiemi (o equivalentemente, funzioni totali caratteristiche). Possiamo estendere a **funzioni parziali**.

Definizione: Per funzioni parziali f, g: $\mathbb{N} \rightharpoonup \mathbb{N}$, definiamo: f ≤_T g se f è computabile con oracolo per g (dove "oracolo per g" significa poter consultare g su qualsiasi input nel dominio di g).

Problema: La relazione ≡_T per funzioni parziali non è transitiva! Controesempio:

- Siano f, g, h funzioni parziali opportune
- Potrebbe valere f ≡_T g e g ≡_T h ma f $\not\equiv$_T h

Soluzione - Gradi di Enumerazione di Rogers:

Definizione (Rogers): Per funzioni parziali f, g, definiamo: f ≤_μ g se il grafico di f è enumerabile dato il grafico di g

Proprietà: ≡_μ è una equivalenza, e i **μ-gradi** formano una struttura ben comportata.

Teorema (Rogers, 1967): La struttura dei μ-gradi è un upper semilattice che estende naturalmente i gradi e-riduttivi.

s-Degrees (Selman)

Definizione: Per linguaggi in complessità limitata (es. NP), due linguaggi A, B hanno lo stesso **s-degree** se: A ≡^P_m B (sono riducibili many-one polinomialmente in entrambe le direzioni)

Applicazione: Classificare problemi NP-completi per "tipo di difficoltà".

Teorema (Selman, 1978): Esistono infiniti s-degrees in NP (assumendo P ≠ NP).

Continuous Degrees

Contesto: Computable analysis - estensione della teoria della computabilità ai numeri reali.

Definizione: Per funzioni continue f, g: [0,1] → $\mathbb{R}$, definiamo f ≤_c g se f è computabile da g (nel senso di type-2 computability).

Struttura: I gradi continui formano un ordine parziale che generalizza i gradi discreti classici.

Teorema (Weihrauch, Brattka): La struttura dei gradi continui ha proprietà topologiche che riflettono la struttura degli spazi metrici sottostanti.

Degree Classes e Transfer Principles

Classi Definibili di Gradi

Definizione: Una **classe definibile** $C \subseteq D$ è un insieme di gradi definibile da una formula del primo ordine in $(D, \leq, \vee, ')$.

Esempi:

- Gradi ricorsivi: {a : a = 0}
- Gradi r.e.: definibile (teorema di Slaman-Soare)
- Gradi low: {a : a' = 0'}
- Gradi high: {a : a' = 0''}

Teorema (Shore, Slaman): La classe dei gradi r.e. è l'unica classe Σ_3-definibile che:

1. Contiene 0
2. È chiusa per join
3. Contiene 0' come elemento massimale

Questo fornisce una caratterizzazione intrinseca dei gradi r.e.

Transfer Principles

Principio: Proprietà vere in D_ω (gradi classici) si trasferiscono a D_α per ordinali ammissibili α?

Teorema (Shore, 1981): Molte proprietà si trasferiscono, ma non tutte. Controesempi:

- La congettura di Slaman-Woodin potrebbe valere in D ma fallire in D_{ω₁^{CK}}
- Proprietà di densità si trasferiscono
- Proprietà di embedding di strutture finite si trasferiscono

Applicazione: Capire quali proprietà sono "accidentali" (specifiche di ω) vs "essenziali" (valgono in ogni livello della gerarchia).

Borel Determinacy e Gradi

Teorema (Martin): Tutti i giochi Borel sono determinati.

Versione Computazionale: La complessità della strategia vincente è misurata da gradi di Turing.

Teorema (Martin, 1975): Per giochi con payoff Σ^0_n:

- Esiste una strategia vincente
- La strategia ha grado ≤ 0^{(n)}

Estensione (Steel, Martin): Per payoff analitici (Σ^1_1), le strategie possono avere gradi iperaritmetici.

Connessione con Large Cardinals: Determinacy per payoff più complessi (proiettivi) richiede assiomi di grande cardinale, collegando teoria dei gradi con teoria degli insiemi profonda.

Conclusioni Metodologiche

La proliferazione di nozioni di riducibilità (Turing, tt, wtt, btt, e, m, Q, Medvedev, Muchnik, h, bt, Lipschitz, generic, μ, s, continuous, ...) non è mera complessificazione arbitraria. Ciascuna nozione cattura un aspetto diverso della **complessità computazionale**:

- **Turing**: Complessità "intrinseca" senza restrizioni
- **tt**: Complessità senza adattività
- **wtt/bt**: Complessità con limiti quantitativi
- **e**: Complessità positiva/monotona
- **h**: Complessità transfinita
- **Medvedev/Muchnik**: Complessità di problemi vs soluzioni individuali
- **s-degrees**: Complessità polinomiale
- **Continuous**: Complessità topologica

Questa **pluralità di prospettive** rivela che la "computabilità" non è un monolite, ma un fenomeno sfaccettato che si manifesta diversamente secondo il contesto matematico. La teoria dei gradi, in tutte le sue varianti, fornisce gli strumenti per navigare questa complessità con precisione matematica rigorosa.